

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Λειτουργικά Συστήματα

Ενότητα # 6: Αδιέξοδα

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



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

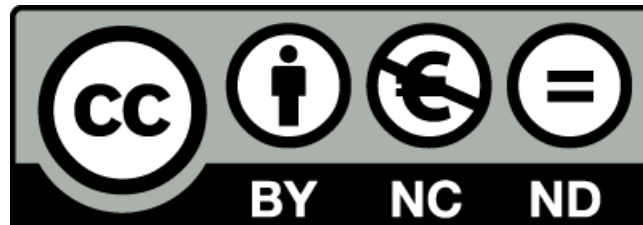
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
πρόγραμμα για την ανάπτυξη

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Οι εικόνες προέρχονται από το βιβλίο «Σύγχρονα Λειτουργικά Συστήματα», A.S. Tanenbaum, 3^η έκδοση, 2009, Εκδόσεις Κλειδάριθμος.



Σκοποί ενότητας

- Κατανόηση της έννοιας των πόρων και του προβλήματος των αδιεξόδων κατά τη χρήση των πόρων
- Εξοικείωση με τις βασικές τεχνικές διαχείρισης αδιεξόδων: αγνόηση, εντοπισμός και ανάκαμψη, αποφυγή, αποτροπή
- Κατανόηση των πρακτικών προβλημάτων στη διαχείριση αδιεξόδων

Περιεχόμενα ενότητας

- Πόροι
- Αδιέξοδα
- Αντιμετώπιση αδιεξόδων
- Εντοπισμός και ανάκαμψη
- Αποφυγή αδιεξόδων
- Αποτροπή αδιεξόδων
- Άλλα θέματα

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Πόροι

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 6:** Αδιέξοδα

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

Τύποι πόρων (1 από 3)

- Ορισμένοι πόροι του ΛΣ δεν είναι κοινόχρηστοι
 - Πόροι υλικού: εκτυπωτής
 - Πόροι λογισμικού: πίνακας διεργασιών
- Η δέσμευση πόρων μπορεί να δώσει αδιέξοδο
 - Η A δεσμεύει το σαρωτή και ζητά τον εκτυπωτή
 - Η B δεσμεύει τον εκτυπωτή και ζητά το σαρωτή
- Τα αδιέξοδα είναι πολύ πιο γενικό πρόβλημα
 - Συμβαίνουν με πολλές διεργασίες (κυκλική αναμονή)
 - Συμβαίνουν σε βάσεις (κλειδώματα εγγραφών)

Τύποι πόρων (2 από 3)

- Πόρος (resource)
 - Γενικός όρος για οντότητες που δεσμεύουμε
 - Πόροι υλικού (ταινίες) ή πληροφοριών (βάση)
 - Μπορούμε να έχουμε πολλούς πόρους ενός τύπου
- Προεκτοπίσιμοι πόροι: αποσπώνται εύκολα
 - Παράδειγμα: μνήμη
 - Η διεργασία A δεσμεύει όλη τη μνήμη και τον εκτυπωτή
 - Η B δεσμεύει όλη τη μνήμη αλλά όχι τον εκτυπωτή
 - Δεν έχουμε αδιέξοδο: παίρνουμε τη μνήμη της B

Τύποι πόρων (3 από 3)

- Μη προεκτοπίσιμοι πόροι: δεν αποσπώνται εύκολα
 - Παράδειγμα: εγγραφές CD
- Τα αδιέξοδα αφορούν *μη προεκτοπίσιμους* πόρους
- Ο τρόπος δέσμευσης εξαρτάται από το ΛΣ
 - Δέσμευση πόρου με κλήση συστήματος request
 - Δέσμευση πόρου με άνοιγμα αρχείου συσκευής
- Τι γίνεται όταν ένας πόρος δεν είναι διαθέσιμος;
 - Είτε η διεργασία μπλοκάρεται και περιμένει
 - Είτε επιστρέφει σφάλμα και ξαναπροσπαθεί

Απόκτηση πόρων (1 από 2)

```
typedef int semaphore;  
semaphore resource_1;
```

```
void process_A(void) {  
    down(&resource_1);  
    use_resource_1();  
    up(&resource_1);  
}
```

(α)

```
typedef int semaphore;  
semaphore resource_1;  
semaphore resource_2;
```

```
void process_A(void) {  
    down(&resource_1);  
    down(&resource_2);  
    use_both_resources();  
    up(&resource_2);  
    up(&resource_1);  
}
```

(β)

- Δέσμευση επιπέδου χρήστη (π.χ. για βάσεις)
 - Χρήση δυαδικού σηματοφόρο με αρχική τιμή 1 (α)
 - Εναλλακτικά, μεταβλητή τύπου mutex
 - Λειτουργεί και με δύο ή περισσότερους πόρους (β)
 - Αποδέσμευση αντίστροφα από τη δέσμευση

Απόκτηση πόρων (2 από 2)

```
typedef int semaphore;  
semaphore resource_1;  
semaphore resource_2;
```

```
void process_A(void) {  
    down(&resource_1);  
    down(&resource_2);  
    use_both_resources();  
    up(&resource_2);  
    up(&resource_1);  
}
```

```
void process_B(void) {  
    down(&resource_1);  
    down(&resource_2);  
    use_both_resources();  
    up(&resource_2);  
    up(&resource_1);  
}
```

(α)

```
typedef int semaphore;  
semaphore resource_1;  
semaphore resource_2;
```

```
void process_A(void) {  
    down(&resource_1);  
    down(&resource_2);  
    use_both_resources();  
    up(&resource_2);  
    up(&resource_1);  
}
```

```
void process_B(void) {  
    down(&resource_2);  
    down(&resource_1);  
    use_both_resources();  
    up(&resource_1);  
    up(&resource_2);  
}
```

(β)

- Πρέπει όλοι να συμφωνούν στη σειρά (α)
 - Αλλιώς, έχουμε κίνδυνο αδιεξόδου (β)

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**

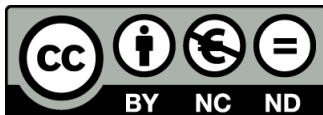


**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Αδιέξοδα

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 6:** Αδιέξοδα

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

Ορισμός αδιεξόδου (1 από 2)

- Τυπικός ορισμός αδιεξόδου
 - Έστω ένα σύνολο μπλοκαρισμένων διεργασιών
 - Όλες οι διεργασίες περιμένουν κάποιο συμβάν
 - Τα συμβάντα αυτά παράγονται μόνο εσωτερικά
 - Αφού όλες περιμένουν κάποια άλλη του συνόλου...
 - ..καμία δεν μπορεί να προχωρήσει!
 - Με ορισμένες υποθέσεις
 - Υποθέτουμε ένα νήμα ανά διεργασία
 - Υποθέτουμε ότι δεν αφυπνίζονται από διακοπές

Ορισμός αδιεξόδου (2 από 2)

- Αδιέξοδο πόρων
 - Σύνολο μπλοκαρισμένων διεργασιών
 - Περιμένουν να απελευθερωθεί κάποιος πόρος
 - Οι πόροι κατέχονται από αυτές τις διεργασίες
 - Ανεξάρτητα από πλήθος και είδος πόρων
 - Η πιο συνηθισμένη μορφή αδιεξόδου
 - Όχι όμως και η μοναδική

Συνθήκες του Coffman (1 από 3)

- Συνθήκες του Coffman
 - Πρέπει να ισχύουν όλες για αδιέξοδο πόρων
 - 1. Αμοιβαίος αποκλεισμός
 - Κάθε πόρος είτε είναι εκχωρημένος σε μία διεργασία είτε είναι διαθέσιμος
 - Δεν επιτρέπεται καταμερισμός των πόρων
 - 2. Δέσμευση και αναμονή
 - Μια διεργασία με εκχωρημένους πόρους μπορεί να ζητάει κι άλλους
 - Δεν ζητάει υποχρεωτικά όλους τους πόρους μαζί

Συνθήκες του Coffman (2 από 3)

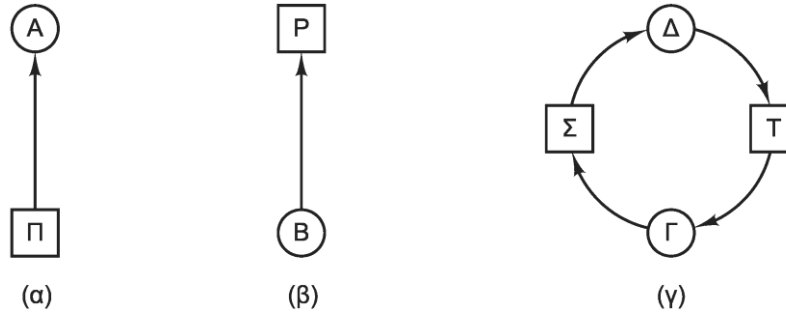
3. Μη προεκτόπιση

- Δεν μπορούμε να αφαιρέσουμε εκχωρημένους πόρους

4. Κυκλική αναμονή

- Κυκλική αλυσίδα διεργασιών που περιμένουν η μία την άλλη
- Κάθε συνθήκη εκφράζει μία πολιτική συστήματος
- Ένα σύστημα μπορεί να την εφαρμόζει ή όχι
- Αρκεί να μην ισχύει μία από αυτές
- Τότε είναι αδύνατον να έχουμε αδιέξοδα

Μοντελοποίηση αδιεξόδων (1 από 3)



- Μοντελοποίηση αδιεξόδων με γράφους
 - Διεργασίες: κύκλοι, πόροι: τετράγωνα
 - Τόξο από πόρο σε διεργασία: κατοχή πόρου
 - Τόξο από διεργασία σε πόρο: αίτηση για πόρο
 - Κύκλος στο γράφο σημαίνει αδιέξοδο
 - Παράδειγμα
 - Η διεργασία Δ κατέχει τον πόρο Σ και ζητάει τον πόρο Τ
 - Η διεργασία Γ κατέχει τον πόρο Τ και ζητάει τον πόρο Σ

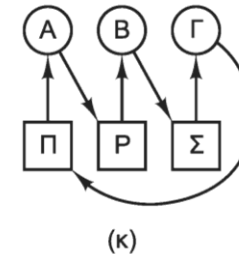
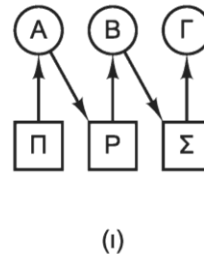
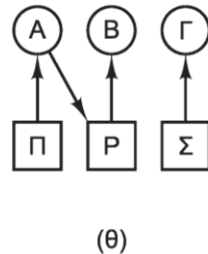
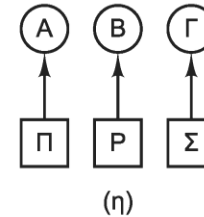
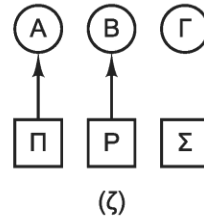
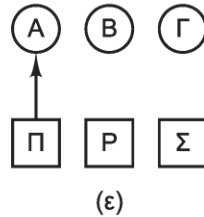
Μοντελοποίηση αδιεξόδων (2 από 3)

A
 Ζητά τον Π
 Ζητά τον Ρ
 Αποδεσμεύει τον Π
 Αποδεσμεύει τον Ρ
 (α)

B
 Ζητά τον Ρ
 Ζητά τον Σ
 Αποδεσμεύει τον Ρ
 Αποδεσμεύει τον Σ
 (β)

Γ
 Ζητά τον Σ
 Ζητά τον Π
 Αποδεσμεύει τον Σ
 Αποδεσμεύει τον Π
 (γ)

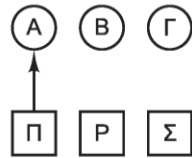
1. Η Α ζητά τον Π
 2. Η Β ζητά τον Ρ
 3. Η Γ ζητά τον Σ
 4. Η Α ζητά τον Ρ
 5. Η Β ζητά τον Σ
 6. Η Γ ζητά τον Π
- Εμφανίζεται αδιέξοδο
 (δ)



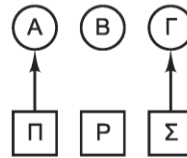
- Παράδειγμα χρήσης γράφου διεργασιών και πόρων
 - Αν εκτελεστούν ακολουθιακά, δεν έχουμε πρόβλημα
 - Αν εκτελεστούν εκ περιτροπής όμως, έχουμε πρόβλημα

Μοντελοποίηση αδιεξόδων (3 από 3)

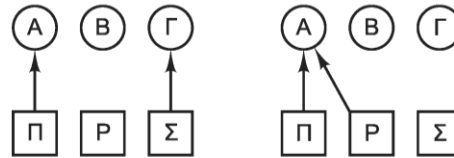
1. Η Α ζητά τον Π
 2. Η Γ ζητά τον Σ
 3. Η Α ζητά τον Ρ
 4. Η Γ ζητά τον Π
 5. Η Α αποδεσμεύει τον Π
 6. Η Α αποδεσμεύει τον Ρ
- Δεν εμφανίζεται αδιέξοδο



(λ)

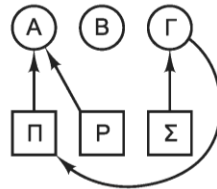


(μ)

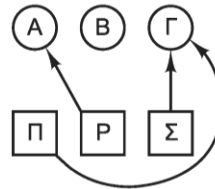


(ν)

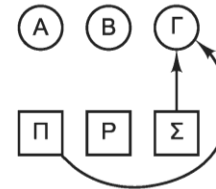
(ξ)



(ο)



(π)



(ρ)

- Παράδειγμα χρήσης γράφου διεργασιών και πόρων
 - Το ΛΣ μπορεί να εκτελέσει εκ περιτροπής τις Α και Γ
 - Η Β θα εκτελεστεί σε επόμενο στάδιο
 - Ο γράφος μας δείχνει αν κινδυνεύουμε από αδιέξοδο

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**

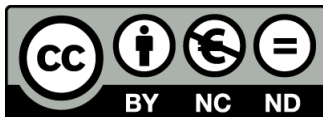


**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Αντιμετώπιση αδιεξόδων

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 6:** Αδιέξοδα

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



Αγνόηση αδιεξόδων (1 από 2)

- Τέσσερις γενικές στρατηγικές αντιμετώπισης
 - Αγνοούμε το πρόβλημα
 - Ελπίζουμε ότι δεν θα συμβούν
 - Εντοπίζουμε τα αδιέξοδα και ανακάμπουμε
 - Απαιτεί ακύρωση ενεργειών και σπατάλη πόρων
 - Αποφεύγουμε τα αδιέξοδα
 - Απαιτεί προσεκτική κατανομή των πόρων
 - Αποτρέπουμε τα αδιέξοδα
 - Άρση των συνθηκών του Coffman

Αγνόηση αδιεξόδων (2 από 2)

- Ο αλγόριθμος της στρουθοκαμήλου
 - Απλά αγνοούμε το πρόβλημα των αδιεξόδων
 - Θεωρητικά, απαράδεκτη προσέγγιση
 - Ρεαλιστικά, έχουμε έναν συμβιβασμό:
 - Πόσο συχνά συμβαίνουν αδιέξοδα;
 - Πόσο κοστίζει στο σύστημα η αντιμετώπιση;
 - Πολλά συστήματα θέλουν τακτική επανεκκίνηση
 - Σφάλματα ΛΣ, αστοχίες υλικού, σφάλματα εφαρμογών
 - Δεν έχει νόημα να ασχολούμαστε ειδικά με τα αδιέξοδα

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**

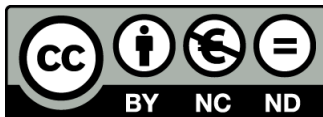


**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Εντοπισμός και ανάκαμψη

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 6:** Αδιέξοδα

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



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

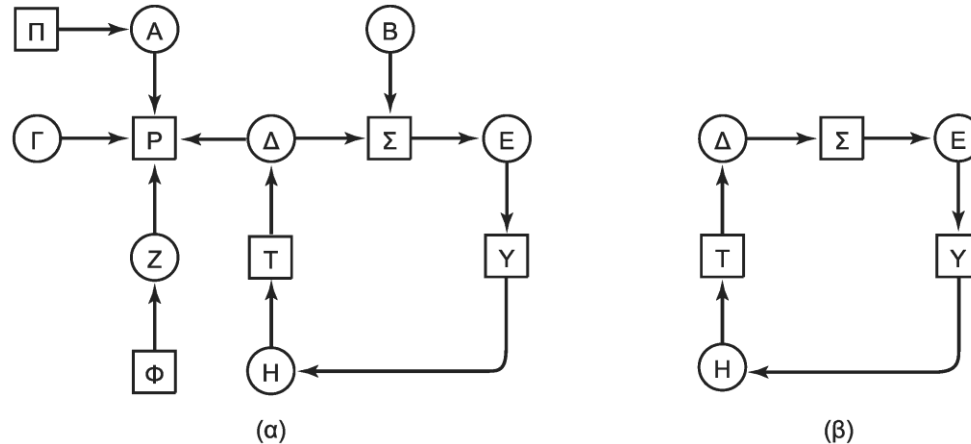


ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Με έναν πόρο (1 από 3)



- Εντοπισμός αδιεξόδων με έναν πόρο ανά είδος
 - Παράδειγμα: 7 διεργασίες (A-H) και 6 πόροι (Π-Φ)
 - Η A κατέχει τον Π και ζητάει τον P, η B ζητάει τον Σ
 - Η Γ ζητάει τον P, η Δ κατέχει τον Τ και ζητάει τους P και Σ
 - Η Ε κατέχει τον Σ και ζητάει τον Υ, η Ζ κατέχει τον Φ και ζητάει τον P
 - Η Η κατέχει τον Υ και ζητάει τον Τ
 - Κατασκευάζουμε τον γράφο (α) και εντοπίζουμε τον κύκλο (β)

Με έναν πόρο (2 από 3)

- Για κάθε κόμβο K εκτελούμε τα επόμενα βήματα από τον K
 1. Αρχικά $\Lambda =$ κενή λίστα και όλα τα τόξα είναι ασημείωτα
 2. Προσθέτουμε τον τρέχοντα κόμβο στο τέλος της Λ
 3. Αν ο κόμβος εμφανίζεται δύο φορές στη Λ , η Λ περιέχει κύκλο
 4. Αν δεν υπάρχουν ασημείωτα τόξα από τον κόμβο πάμε στο 7

Με έναν πόρο (3 από 3)

5. Επιλέγουμε και σημειώνουμε ένα ασημείωτο εξερχόμενο τόξο
6. Χρησιμοποιούμε τον νέο κόμβο ως τρέχοντα και πάμε στο 2
7. Αν ο τρέχων κόμβος είναι ο αρχικός, τέλος χωρίς κύκλο
8. Αλλιώς, αφαιρούμε τον κόμβο από τη λίστα
9. Κάνουμε τον προηγούμενο της λίστας τρέχοντα και πάμε στο 2

Με πολλούς πόρους (1 από 4)

Υπάρχοντες πόροι
($Y_1, Y_2, Y_3, \dots, Y_\mu$)

Διαθέσιμοι πόροι
($\Theta_1, \Theta_2, \Theta_3, \dots, \Theta_\mu$)

Μητρώο τρέχουσας κατανομής

$$\begin{bmatrix} T_{11} & T_{12} & T_{13} & \cdots & T_{1\mu} \\ T_{21} & T_{22} & T_{23} & \cdots & T_{2\mu} \\ \vdots & \vdots & \vdots & & \vdots \\ T_{v1} & T_{v2} & T_{v3} & \cdots & T_{v\mu} \end{bmatrix}$$

Η v -οστή γραμμή δείχνει πόσα αντίγραφα κάθε πόρου είναι δεσμευμένα από τη διεργασία v

Μητρώο αιτήσεων

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1\mu} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2\mu} \\ \vdots & \vdots & \vdots & & \vdots \\ A_{v1} & A_{v2} & A_{v3} & \cdots & A_{v\mu} \end{bmatrix}$$

Η γραμμή 2 δείχνει τους πόρους που χρειάζεται η διεργασία 2

- Εντοπισμός αδιεξόδων με πολλούς πόρους ανά είδος
 - Το διάνυσμα υπαρχόντων Y δείχνει πόσοι υπάρχουν
 - Το διάνυσμα διαθέσιμων Θ δείχνει πόσοι είναι διαθέσιμοι
 - Το μητρώο κατανομής T δείχνει πού εκχωρήθηκαν
 - Το μητρώο αιτήσεων A δείχνει πόσοι μπορεί να ζητηθούν

Με πολλούς πόρους (2 από 4)

$$\sum_{k=1}^n T_{kl} + \Theta_l = Y_l$$

- Βασική συνθήκη πόρων
 - Κατανεμημένοι και διαθέσιμοι = υπάρχοντες
 - Οι αιτήσεις δείχνουν πόσοι ακόμη χρειάζονται
- Υποθέσεις αλγόριθμου εντοπισμού αδιεξόδων
 - $A \leq B$ αν όλα τα στοιχεία του $A \leq$ των αντίστοιχων του B
 - Κάθε διεργασία διατηρεί τους πόρους της μέχρι τέλους
 - Αρχικά όλες οι διεργασίες είναι ασημείωτες
 - Όσες παραμείνουν ασημείωτες βρίσκονται σε αδιέξοδο

Με πολλούς πόρους (3 από 4)

- Αλγόριθμος εντοπισμού αδιεξόδων
 1. Ψάξε μία ασημείωτη διεργασία k με $A_k \leq \Theta$
 2. Αν βρεθεί τέτοια, πρόσθεσε το T_k στο Θ
 3. Μετά σημείωσε τη διεργασία και πήγαινε στο 1
 4. Αν δεν υπάρχει, ο αλγόριθμος τερματίζεται
- Ψάχνουμε μια διεργασία που μπορεί να ολοκληρωθεί
 - Θεωρούμε ότι δεσμεύει όλους τους πόρους που θέλει
 - Μετά εκτελείται ως το τέλος και τους απελευθερώνει
 - Άρα ψάχνουμε μια ασφαλή ακολουθία εκτέλεσης

Με πολλούς πόρους (4 από 4)

Μονάδες ταινίας	Σχεδιογράφοι	Σαρωτές	Μονάδες CD ROM
Υ = (4 2 3 1)	Θ = (2 1 0 0)		
Μητρώο τρέχουσας κατανομής	Μητρώο αιτήσεων		
$T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \end{bmatrix}$	$A = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{bmatrix}$		

- Παράδειγμα αλγόριθμου εντοπισμού αδιεξόδων
 - Πρώτα μπορεί να εκτελεστεί η διεργασία 3
 - Υπόλοιπο πόρων (2,2,2,0)
 - Μετά μπορεί να εκτελεστεί η διεργασία 2
 - Υπόλοιπο πόρων (4,2,2,1)
 - Τέλος μπορεί να εκτελεστεί η διεργασία 1

Ανάκαμψη από αδιέξοδα (1 από 4)

- Πότε ελέγχουμε για αδιέξοδα;
 - Είτε όποτε ζητείται κάποιος πόρος
 - Μεγάλη επιβάρυνση
 - Είτε περιοδικά
 - Κάθε k χτύπους του ρολογιού
 - Είτε όταν ο φόρτος πέσει πολύ
 - Μπορεί να οφείλεται σε αδιέξοδο

Ανάκαμψη από αδιέξοδα (2 από 4)

- Ανάκαμψη μέσω προεκτόπισης
 - Παράδειγμα: αδιέξοδο που εμπλέκει τον εκτυπωτή
 - Διακόπτουμε τη διεργασία που έχει τον εκτυπωτή
 - Μαζεύουμε την έξοδό της και παραχωρούμε τον εκτυπωτή
 - Όταν τελειώσει, ενεργοποιούμε ξανά την αρχική διεργασία
 - Γενικά αυτό δεν είναι πάντα εφικτό
 - Παράδειγμα: εγγραφέας CD που γράφει ένα CD-R

Ανάκαμψη από αδιέξοδα (3 από 4)

- Ανάκαμψη μέσω ανασκευής (rollback)
 - Περιοδική δημιουργία σημείων ελέγχου
 - Λέγονται και checkpoints
 - Εγγραφή κατάστασης διεργασίας σε αρχείο
 - Περιλαμβάνει εικόνα μνήμης και πόρους
 - Εντοπίζουμε τη διεργασία που εμπλέκεται
 - Εντοπίζουμε τον πόρο που έχει πρόβλημα
 - Επανεκκίνηση από προηγούμενο σημείο ελέγχου
 - Πρέπει να μην έχει δεσμευθεί ακόμα ο πόρος

Ανάκαμψη από αδιέξοδα (4 από 4)

- Ανάκαμψη μέσω εξάλειψης διεργασιών
 - Σκοτώνουμε μία διεργασία του κύκλου
 - Ελπίζουμε ότι ο κύκλος θα σπάσει
 - Σκοτώνουμε μία διεργασία με κατάλληλους πόρους
 - Αυτούς που ζητούνται στον κύκλο
 - Ο στόχος είναι να σπάσει ο κύκλος αναμονής
 - Προτιμάμε διεργασίες που τερματίζονται εύκολα
 - Κατάλληλη: μεταγλώττιση
 - Ακατάλληλη: βάση δεδομένων

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Αποφυγή αδιεξόδων

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 6:** Αδιέξοδα

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

Γιατί αποφυγή αδιεξόδων;

- Είναι ρεαλιστικός ο εντοπισμός αδιεξόδων;
 - Απαιτεί γνώση που σπάνια έχουμε
 - Πρέπει να γνωρίζουμε τις απαιτήσεις πόρων!
 - Είναι πολύ συντηρητικός
 - Υποθέτει ότι η διεργασία θα τους ζητήσει όλους μαζί
 - Χρειαζόμαστε πιο ρεαλιστική προσέγγιση

Τροχιές πόρων (1 από 2)

- Τροχιές πόρων
 - Έστω ότι έχουμε δύο διεργασίες και δύο πόρους
 - Κάθε διεργασία χρειάζεται τους πόρους για κάποιο διάστημα
 - Άξονας: εξέλιξη μίας διεργασίας στο χρόνο
 - Η τεθλασμένη δείχνει την εξέλιξη των διεργασιών
 - Κίνηση μόνο προς τα πάνω ή τα δεξιά

Ασφαλείς καταστάσεις (1 από 2)

	Κατέχει	Μέγιστο		Κατέχει	Μέγιστο		Κατέχει	Μέγιστο		Κατέχει	Μέγιστο		Κατέχει	Μέγιστο
A	3	9	A	3	9	A	3	9	A	3	9	A	3	9
B	2	4	B	4	4	B	0	-	B	0	-	B	0	-
Γ	2	7	Γ	2	7	Γ	2	7	Γ	7	7	Γ	0	-
Ελεύθεροι: 3		Ελεύθεροι: 1		Ελεύθεροι: 5		Ελεύθεροι: 0		Ελεύθεροι: 7						
(α)		(β)		(γ)		(δ)		(ε)						

- Ασφαλείς και ανασφαλείς καταστάσεις
 - Ίδια κατάσταση με τον αλγόριθμο εντοπισμού
 - Διανύσματα υπαρχόντων Υ και διαθέσιμων Θ
 - Μητρώα κατανομής T και αιτήσεων A
 - Μία κατάσταση είναι ασφαλής (safe) εάν
 - Δεν είμαστε σε αδιέξοδο
 - Υπάρχει κάποια σειρά ολοκλήρωσης των διεργασιών
 - Παράδειγμα με έναν πόρο και τρεις διεργασίες
 - Αρχικά εκτελείται η B και μετά η Γ μέχρι τέλους

Ασφαλείς καταστάσεις (2 από 2)

Κατέχει Μέγιστο

A	3	9
B	2	4
Γ	2	7

Ελεύθεροι: 3
(α)

Κατέχει Μέγιστο

A	4	9
B	2	4
Γ	2	7

Ελεύθεροι: 2
(β)

Κατέχει Μέγιστο

A	4	9
B	4	4
Γ	2	7

Ελεύθεροι: 0
(γ)

Κατέχει Μέγιστο

A	4	9
B	-	-
Γ	2	7

Ελεύθεροι: 4
(δ)

- Παράδειγμα με έναν πόρο και τρεις διεργασίες
 - Έστω ότι η A ζητάει ένα ακόμη αντίγραφο του πόρου
 - Αρχικά εκτελείται η B μέχρι τέλους
 - Μετά όμως δεν μπορεί να εκτελεστεί ούτε η A ούτε η Γ
 - Άρα η κατάσταση δεν είναι ασφαλής
- Προσοχή: ανασφαλής κατάσταση \leftrightarrow αδιέξοδο
 - Οι διεργασίες δεν ζητάνε απαραίτητα όλους τους πόρους
 - Απλά δεν είναι εγγυημένο ότι δεν θα έχουμε αδιέξοδο

Αλγόριθμος τραπεζίτη (1 από 3)

Μέγιστο
Κατέχει

A	0	6
B	0	5
Γ	0	4
Δ	0	7

Ελεύθεροι: 10

(α)

Μέγιστο
Κατέχει

A	1	6
B	1	5
Γ	2	4
Δ	4	7

Ελεύθεροι: 2

(β)

Μέγιστο
Κατέχει

A	1	6
B	2	5
Γ	2	4
Δ	4	7

Ελεύθεροι: 1

(γ)

- Ο αλγόριθμος του τραπεζίτη για έναν πόρο
 - Επέκταση του αλγορίθμου εντοπισμού (Dijkstra)
 - Πριν δώσουμε πόρο, βλέπουμε αν πάμε σε ασφαλής κατάσταση
 - Αν όχι, τότε δεν παραχωρούμε τον πόρο
 - Παράδειγμα με έναν πόρο
 - (α) αρχική κατάσταση με μέγιστες απαιτήσεις
 - (β) ασφαλής κατάσταση (σειρά ολοκλήρωσης Γ, Δ, Β, Α)
 - (γ) ανασφαλής κατάσταση (δεν υπάρχει σειρά ολοκλήρωσης)

Αλγόριθμος τραπεζίτη (2 από 3)

- Ο αλγόριθμος του τραπεζίτη για πολλούς πόρους
 - Μητρώο εκχώρησης και πρόσθετων αιτήσεων
 - Διανύσματα πόρων (Y), εκχωρημένων (K), διαθεσίμων (Θ)
- 1. Βρες μία διεργασία R που μπορεί να ολοκληρωθεί
 - Η γραμμή της R πρέπει να είναι μικρότερη ή ίση του Θ
 - Θεωρούμε ότι δεσμεύονται οι πόροι και μετά τερματίζει η R
- 2. Απελευθέρωσε τους πόρους προσθέτοντάς τους στο Θ
- 3. Επανάλαβε τα βήματα 1 και 2
 - Είτε τερματίζουν όλες οι διεργασίες
 - Είτε η κατάσταση ήταν ανασφαλής

Αλγόριθμος τραπεζίτη (3 από 3)

Διεργασία	Μονάδες ταινίας	Σχεδιογράφοι	Εκτυπωτές	Μονάδες CD-ROM
A	3	0	1	1
B	0	1	0	0
Γ	1	1	1	0
Δ	1	1	0	1
E	0	0	0	0

Πόροι που έχουν εκχωρηθεί

Διεργασία	Μονάδες ταινίας	Σχεδιογράφοι	Εκτυπωτές	Μονάδες CD-ROM
A	1	1	0	0
B	0	1	1	2
Γ	3	1	0	0
Δ	0	0	1	0
E	2	1	1	0

Πόροι που χρειάζονται ακόμη

Y = (6342)
K = (5322)
Θ = (1020)

- Ο αλγόριθμος του τραπεζίτη για πολλούς πόρους
 - Στο παράδειγμα η κατάσταση είναι ασφαλής
 - Σειρά τερματισμού Δ, A, E, ...
 - Έστω ότι η B ζητάει έναν εκτυπωτή
 - Πάλι σειρά τερματισμού Δ, A, E, ...
 - Έστω ότι η E ζητάει τον τελευταίο εκτυπωτή
 - Η κατάσταση είναι ανασφαλής

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**

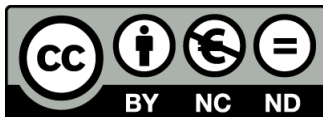


**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Αποτροπή αδιεξόδων

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 6:** Αδιέξοδα

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 πρόληψη αδιεξόδων

- Πόσο ρεαλιστική είναι η αποφυγή αδιεξόδων;
 - Λίγο καλύτερη από τον εντοπισμό
 - Ελέγχουμε τους πόρους μόνο όταν ζητούνται
 - Οι υποθέσεις όμως πάλι δεν είναι ρεαλιστικές
 - Οι απαιτήσεις των διεργασιών δεν είναι γνωστές
 - Νέες διεργασίες μπορεί να έρχονται στο σύστημα

Προσβολή των συνθηκών

- Προσβολή των συνθηκών του Coffman
 - Φροντίζουμε τα αδιέξοδα να είναι αδύνατα
 - Αρκεί να μην ισχύει ποτέ μία συνθήκη του Coffman
 - Πρέπει να ισχύουν όλες για να έχουμε αδιέξοδα
 - Πιο πρακτική λύση σε σχέση με την αποφυγή
 - Απαιτεί όμως επιβολή περιορισμών στο σύστημα

Συνθήκη αμοιβαίου αποκλεισμού

- Προσβολή συνθήκης αμοιβαίου αποκλεισμού
 - Ορισμένοι πόροι απαιτούν αμοιβαίο αποκλεισμό
 - Παράδειγμα: ο εκτυπωτής
 - Κρύβουμε τον εκτυπωτή πίσω από μια διεργασία
 - Μόνο η διεργασία αυτή μπορεί να τυπώσει
 - Οι άλλες διεργασίες της στέλνουν δεδομένα για εκτύπωση
 - Η ίδια η διεργασία δεσμεύει μόνο τον εκτυπωτή
 - Προσοχή όμως για να αποφύγουμε άλλα αδιέξοδα
 - Παράδειγμα: αδιέξοδο στην ουρά εκτύπωσης!

Συνθήκη δέσμευσης και αναμονής

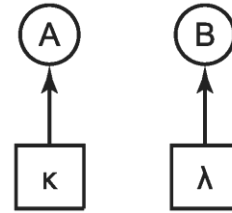
- Προσβολή συνθήκης δέσμευσης και αναμονής
 - Γιατί να μην δεσμεύονται οι πόροι από την αρχή;
 - Είτε η διεργασία θα λάβει όλους τους πόρους
 - Είτε δεν θα είναι διαθέσιμοι και θα περιμένει
 - Πώς μπορούμε να ξέρουμε τι θα χρειαστούμε;
 - Το ίδιο πρόβλημα με την αποφυγή αδιεξόδων
 - Κακή χρήση των πόρων
 - Έστω ότι ξέρουμε τις μέγιστες απαιτήσεις της διεργασίας
 - Δεν χρειάζονται όλοι οι πόροι ταυτόχρονα
 - Δεσμεύουμε πόρους που δεν χρειαζόμαστε

Συνθήκη μη προεκτόπισης

- Προσβολή συνθήκης μη προεκτόπισης
 - Γενικά μη εφικτό σε ορισμένες συσκευές
 - Παράδειγμα: εκτυπωτής (χάνουμε χαρτί)
 - Παράδειγμα: εγγραφέας CD (χάνουμε το CD)
 - Χρήση διαχειριστών πόρων
 - Διαχειριστής εκτυπώσεων
 - Αποκρύπτει την ακολουθιακή χρήση των πόρων
 - Ίδια προσέγγιση με τον αμοιβαίο αποκλεισμό

Συνθήκη κυκλικής αναμονής (1 από 2)

1. Εικονοθέτης
2. Σαρωτής
3. Σχεδιογράφος
4. Μονάδα ταινίας
5. Μονάδα CD-ROM



(α)

(β)

- Προσβολή συνθήκης κυκλικής αναμονής
 - Αρίθμηση όλων των πόρων του συστήματος
 - Οι διεργασίες ζητούν πόρους μόνο με αύξουσα σειρά
 - Παράδειγμα: αν έχεις τον σαρωτή, μπορείς να ζητήσεις ταινία
 - Ουσιαστικά απαγορεύουμε τους κύκλους
 - Κάθε αλυσίδα πόρων περιλαμβάνει πόρους σε αύξουσα σειρά
 - Άρα δεν μπορεί να υπάρξει κύκλος!
 - Περιορίζουμε τον τρόπο λειτουργίας των διεργασιών

Συνθήκη κυκλικής αναμονής (2 από 2)

- Προσβολή συνθήκης κυκλικής αναμονής
 - Παραλλαγή: δεν ζητάς «μικρότερους» πόρους
 - Πρώτα απελευθέρωση «μεγαλύτερων» πόρων
 - Ζητάμε τον σαρωτή όταν αφήσουμε την ταινία
 - Πάλι οι αλυσίδες δεν μπορούν να γίνουν κύκλοι
 - Ποια είναι η σωστή αρίθμηση των πόρων;
 - Δεν είναι καθόλου προφανές!
 - Κάθε πρόγραμμα προτιμάει διαφορετική σειρά
 - Με πολλούς πόρους είναι δύσκολο να βρεθεί σειρά

Σύγκριση

Συνθήκη	Προσέγγιση
Αμοιβαίος αποκλεισμός	Παροχέτευση στα πάντα
Δέσμευση και αναμονή	Να ζητούνται όλοι οι πόροι από την αρχή
Μη προεκτόπιση	Αφαίρεση των πόρων
Κυκλική αναμονή	Αριθμητική διάταξη των πόρων

- Τέσσερις συνθήκες του Coffman
 - Μόνο δύο προσεγγίσεις λειτουργούν καλά
 - Διαχειριστές αμοιβαία αποκλειόμενων πόρων
 - Διατεταγμένη εκχώρηση πόρων
 - Αλλιώς έχουμε σπατάλη πόρων
 - Αφαίρεση ή πλήρης δέσμευση των πόρων

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Άλλα θέματα

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 6:** Αδιέξοδα

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

Κλείδωμα σε δύο φάσεις (1 από 2)

- Κλείδωμα σε δύο φάσεις
 - Χρησιμοποιείται σε βάσεις δεδομένων
 - Φάση 1: προσπαθούμε να κλειδώσουμε τις εγγραφές
 - Αν αποτύχουμε, ξεκλειδώνουμε τα πάντα και ξαναπροσπαθούμε
 - Φάση 2: ενημερώνουμε τις εγγραφές
 - Μετά απελευθερώνουμε όλα τα κλειδώματα

Κλείδωμα σε δύο φάσεις (2 από 2)

- Δέσμευση όλων των πόρων από την αρχή
 - Για κάθε ομάδα ενημερώσεων εγγραφών όμως
- Το κλείδωμα μπορεί να καθυστερήσει πάρα πολύ
- Πρέπει το πρόγραμμα να μπορεί να κάνει πίσω
 - Απαιτείται κατάλληλη σχεδίαση του προγράμματος

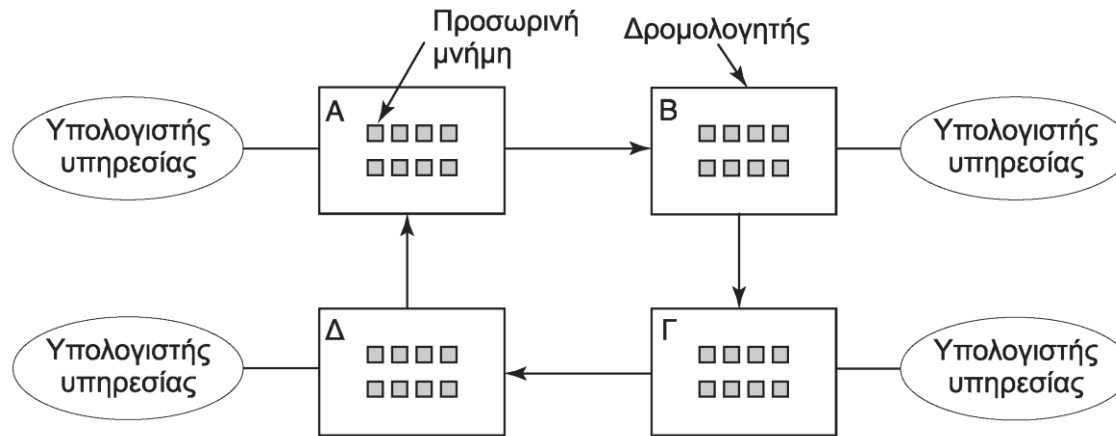
Αδιέξοδα επικοινωνίας (1 από 3)

- Αδιέξοδα επικοινωνίας
 - Η διεργασία A στέλνει ένα μήνυμα στη B
 - Η διεργασία A μπλοκάρει περιμένοντας απάντηση
 - Αν χαθεί η απάντηση η A θα μείνει μπλοκαρισμένη
 - Η B θα περιμένει για ένα νέο μήνυμα από την A
 - Έχουμε αδιέξοδο σύμφωνα με τον συνήθη ορισμό
 - Όλες οι διεργασίες περιμένουν άλλες της ομάδας

Αδιέξοδα επικοινωνίας (2 από 3)

- Η διαχείριση αδιεξόδων πόρων δεν λειτουργεί
- Υπάρχει μια άλλη λύση: τα χρονόμετρα
 - Όταν στέλνεται ένα μήνυμα, ξεκινάει ένα χρονόμετρο
 - Αν λήξει χωρίς απάντηση, επαναλαμβάνουμε
 - Τι θα γίνει όμως αν το μήνυμα απλά άργησε;

Αδιέξοδα επικοινωνίας (3 από 3)



- Αδιέξοδο πόρων σε δίκτυα
 - Οι υπολογιστές υπηρεσίες ανήκουν στους χρήστες
 - Οι κόμβοι του δικτύου είναι δρομολογητές
 - Οι δρομολογητές έχουν πεπερασμένο χώρο αποθήκευσης
 - Στο παράδειγμα έχουν γεμίσει όλοι οι χώροι
 - Κανένας δρομολογητής δεν μπορεί να στείλει μήνυμα!

Ενεργό αδιέξοδο (1 από 2)

```
void process_A(void)
{
    enter_region(&resource_1);
    enter_region(&resource_2);
    use_both_resources();
    leave_region(&resource_2);
    leave_region(&resource_1);
}
```

```
void process_B(void)
{
    enter_region(&resource_2);
    enter_region(&resource_1);
    use_both_resources();
    leave_region(&resource_1);
    leave_region(&resource_2);
}
```

- Ενεργό αδιέξοδο σε κρίσιμες περιοχές
 - Έστω ότι χρησιμοποιούμε αναμονή με απασχόληση
 - Για αποφυγή της επιβάρυνσης από το μπλοκάρισμα
 - Οι δύο διεργασίες προσπαθούν να αποκτήσουν πόρους
 - Με διαφορετική σειρά, οπότε αποκτούν από έναν πόρο
 - Οι διεργασίες εκτελούνται χωρίς να κάνουν πρόοδο
 - Η κατάσταση αυτή λέγεται ενεργό αδιέξοδο (livelock)
 - Δεν έχουμε μπλοκάρισμα, αλλά δεν έχουμε και πρόοδο

Ενεργό αδιέξοδο (2 από 2)

- Πολλοί τρόποι να συμβεί ενεργό αδιέξοδο
- Συνήθως εξαντλείται κάποιος πεπερασμένος πόρος
 - Παράδειγμα: πίνακας διεργασιών
 - Ο πίνακας είναι γεμάτος και τα `fork()` αποτυγχάνουν
 - Παράδειγμα: πίνακας αρχείων
 - Δεν μπορούν να ανοίξουν αρχεία και τα `open()` αποτυγχάνουν
 - Τα περισσότερα συστήματα αγνοούν το πρόβλημα
 - Είναι απίθανο να συμβεί με λογικό φόρτο
 - Αρκεί να έχουμε διαστασιολογήσει σωστά τους πίνακες
 - Οι διεργασίες μπορούν να είναι προετοιμασμένες
 - Αν δεν δεσμευτεί ο πόρος η φορές, αποτυχία και τερματισμός

Λιμοκτονία (1 από 2)

- Ποιος θα πάρει έναν πόρο που ζητούν πολλοί;
 - Κάθε σύστημα εφαρμόζει κάποια πολιτική
- Παράδειγμα: εκτυπωτής
 - Έστω ότι επιλέγουμε τη μικρότερη εκτύπωση
 - Έτσι οι μικρές εκτυπώσεις εξυπηρετούνται γρήγορα
 - Οι μεγάλες όμως μπορεί να μην εκτελεστούν ποτέ!

Λιμοκτονία (2 από 2)

- Η αναβολή εξυπηρέτησης λέγεται λιμοκτονία (starvation)
- Αποφεύγεται με κατάλληλους αλγόριθμους
 - Ο FCFS δεν οδηγεί σε λιμοκτονία
 - Αλλά δεν είναι αποδοτικός!
 - Round Robin (διεργασίες) και SCAN (δίσκος)
 - Δεν οδηγούν σε λιμοκτονία
 - Είναι αποδοτικοί

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**

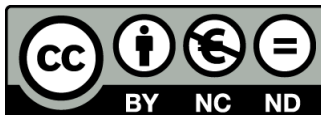


**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Τέλος Ενότητας #6

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 6:** Αδιέξοδα

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

