



ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ ΙΙ (Θ)

Ενότητα 1: ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ ΙΙ

- Νικολαΐδης Αθανάσιος
- Διδάκτορας Ανάπτυξης Τεχνικών Προστασίας Πληροφορίας Εικόνας
- ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Κεντρικής Μακεδονίας» 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.



Ενότητα 1

ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ Ι

Νικολαΐδης Αθανάσιος
Διδάκτορας Ανάπτυξης Τεχνικών
Προστασίας Πληροφορίας Εικόνας

Περιεχόμενα ενότητας

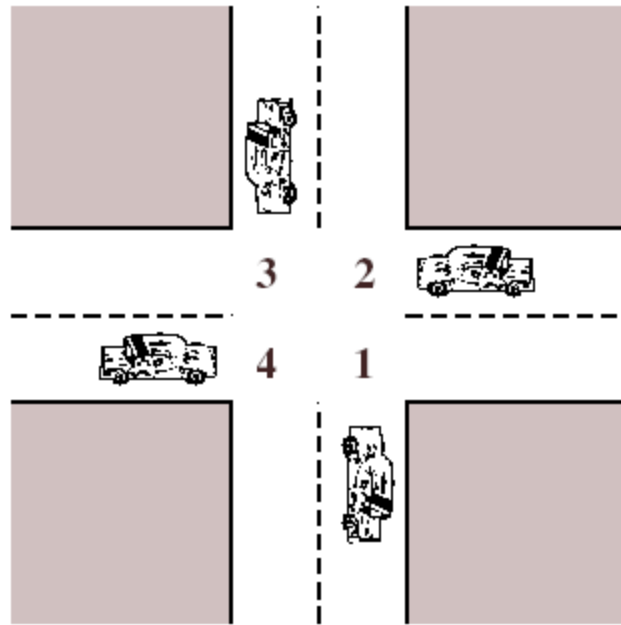
1. Ευρυζωνική Πρόσβαση ΜΕΤΑΛΛΟΥΡΓΙΚΕΣ
2. Ασύρματη Ευρυζωνική Πρόσβαση
3. Ευρυζωνικά Ασύρματα Δίκτυα
4. Ανάπτυξη της Ευρυζωνικότητας
5. Τεχνικές Προκλήσεις Ευρυζωνική Πρόσβαση
6. Τεχνικές Προκλήσεις- Ασύρματο ραδιοκυματικό κανάλι
7. Τεχνικές Προκλήσεις- Παρεμπόδιση λόγω μεγάλων εμποδίων
– Διακύμανση περιβάλλουσας
8. Τεχνικές Προκλήσεις- Διασυμβολική Παρεμβολή
9. Τεχνικές Προκλήσεις- Διασπορά συχνότητας λόγω κίνησης
10. Τεχνικές Προκλήσεις- Παρεμβολή

Σκοποί ενότητας

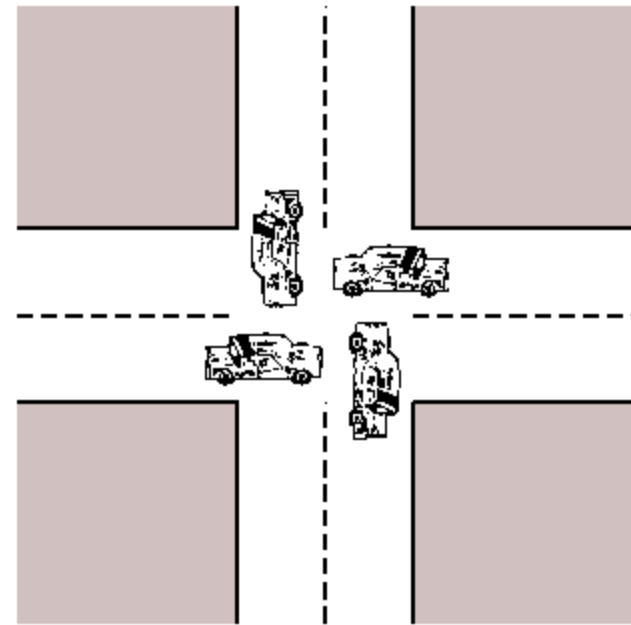
Αδιέξοδο

- Μόνιμο μπλοκάρισμα ενός συνόλου διεργασιών που είτε συναγωνίζονται για πόρους του συστήματος είτε επικοινωνούν μεταξύ τους
- Δεν υπάρχει αποδοτική λύση
- Εμπλέκει συγκρουόμενα συμφέροντα για αγαθά από δύο ή περισσότερες διεργασίες

Παράδειγμα αδιεξόδου



(a) Deadlock possible



(b) Deadlock

Είδη αγαθών

- Προεκτοπίσιμα: μπορούν να αφαιρεθούν από μια διεργασία χωρίς συνέπειες (π.χ. μνήμη)
- Μη προεκτοπίσιμα: η διεργασία αποτυγχάνει αν αφαιρεθεί τέτοιο αγαθό (π.χ. CD-ROM)
- Αδιέξοδα με προεκτοπίσιμα αγαθά επιλύονται εύκολα (ανακατανομή αγαθών)
- Αδιέξοδα με μη προεκτοπίσιμα;

Χρήση αγαθών

- Ακολουθία συμβάντων για χρησιμοποίηση αγαθών από διεργασία:
 - Αίτηση για δέσμευση
 - Χρήση αγαθού
 - Αποδέσμευση αγαθού

Παράδειγμα χωρίς αδιέξοδο

```
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)

Παράδειγμα με αδιέξοδο

```
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)

Τυπικός ορισμός αδιεξόδου

- Ένα σύνολο διεργασιών βρίσκεται σε **αδιέξοδο** αν κάθε διεργασία του συνόλου περιμένει ένα συμβάν που μόνο μια άλλη διεργασία του συνόλου μπορεί να προκαλέσει.

Συνθήκες δημιουργίας αδιεξόδου

- **Αμοιβαίος αποκλεισμός:** μόνο μια διεργασία τη φορά μπορεί να χρησιμοποιήσει ένα αγαθό.
- **Κράτηση και αναμονή:** μια διεργασία που έχει τουλάχιστον ένα αγαθό περιμένει να αποκτήσει επιπλέον άλλα που κατέχουν άλλες διεργασίες.
- **Μη προεκτόπιση:** ένα αγαθό μπορεί να ελευθερωθεί μόνο εθελοντικά από τη διεργασία που το κατέχει.
- **Κυκλική αναμονή:** υπάρχει ένα σύνολο $\{P_0, P_1, P_2, \dots, P_v\}$ από διεργασίες έτσι ώστε η P_0 περιμένει για ένα αγαθό που κατέχει η P_1 , η P_1 περιμένει για ένα αγαθό που κατέχει η P_2 , ... και η P_v περιμένει για ένα αγαθό που κατέχει η P_0 .

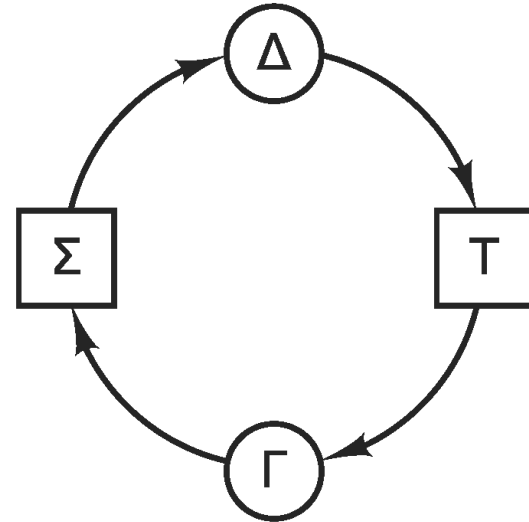
Μοντελοποίηση αδιεξόδων



(α)



(β)



(γ)

Γράφοι κατανομής αγαθών

(α) Η διεργασία A κατέχει τον πόρο Π.

(β) Η διεργασία B ζητάει τον πόρο P.

(γ) Αδιέξοδο.

Τρόποι αντιμετώπισης αδιεξόδων

- Αγνόηση του προβλήματος
- Πρόληψη
- Αποφυγή
- Ανίχνευση και ανάληψη

Πρόληψη αδιεξόδων

- Προσβολή μιας από τις συνθήκες δημιουργίας αδιεξόδου
- Προσβολή της συνθήκης αμοιβαίου αποκλεισμού:
 - Για ορισμένες συσκευές μπορώ να χρησιμοποιήσω spooling
 - Δεν είναι δυνατό για όλες τις συσκευές

Πρόληψη αδιεξόδων

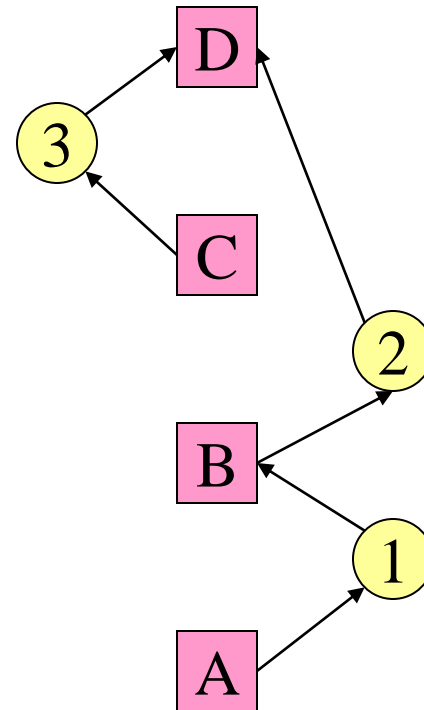
- Προσβολή της συνθήκης κράτησης και αναμονής:
 - Απαίτηση οι διεργασίες να ζητούν από την αρχή ότι θα χρησιμοποιήσουν, όμως...
 - δύσκολο να το γνωρίζει εκ των προτέρων και
 - Δεσμεύει αγαθά που θα μπορούσαν να χρησιμοποιούνται από άλλη διεργασία
 - Παραλλαγή: η διεργασία ελευθερώνει όλα τα αγαθά και τα ξαναζητάει όλα μαζί με το νέο

Πρόληψη αδιεξόδων

- Προσβολή της συνθήκης μη προεκτόπισης:
 - Δύσκολο να εφαρμοστεί γιατί αφήνει τη διεργασία σε ασυνεπή κατάσταση
 - Σε μερικές περιπτώσεις είναι δυνατό (π.χ. αφαίρεση από τη μνήμη κάποιες σελίδες μιας διεργασίας και ανάστειλέ την)

Πρόληψη αδιεξόδων

- Προσβολή της συνθήκης κυκλικής αναμονής:
 - Διάταξη των αγαθών σε ορισμένη σειρά
 - Δέσμευση αγαθών σύμφωνα με τη σειρά αυτή
 - Μια διεργασία που κατέχει το αγαθό n δε μπορεί να περιμένει για το αγαθό m με $m < n$
 - Δε μπορεί να προκύψει κύκλος στο γράφο



Αποφυγή αδιεξόδων

- Λαμβάνεται δυναμικά μια απόφαση αν η νέα αίτηση για εκχώρηση αγαθού μπορεί να οδηγήσει σε αδιέξοδο
- Απαιτείται γνώση μελλοντικών αιτημάτων των διεργασιών

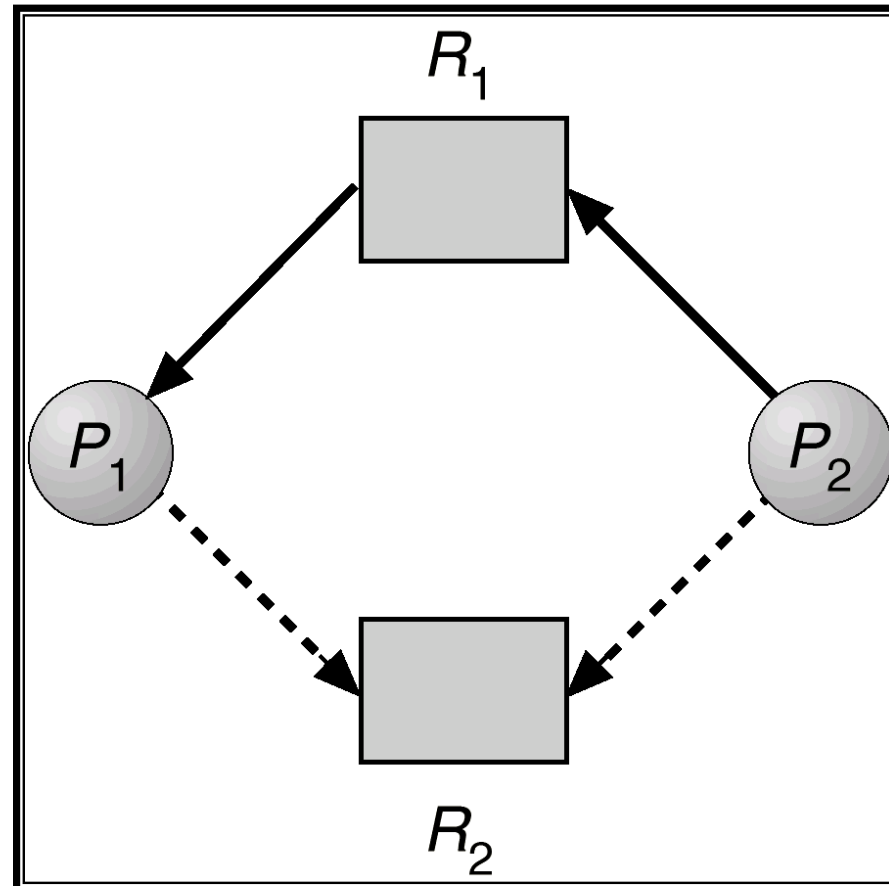
Αποφυγή αδιεξόδων

- Προσέγγιση 1: μην ξεκινάς μια διεργασία αν οι απαιτήσεις της μπορεί να οδηγήσουν σε αδιέξοδο
- Προσέγγιση 2: μην ικανοποιείς ένα νέο αίτημα διεργασίας αν αυτή η εκχώρηση μπορεί να οδηγήσει σε αδιέξοδο

Αποφυγή αδιεξόδων

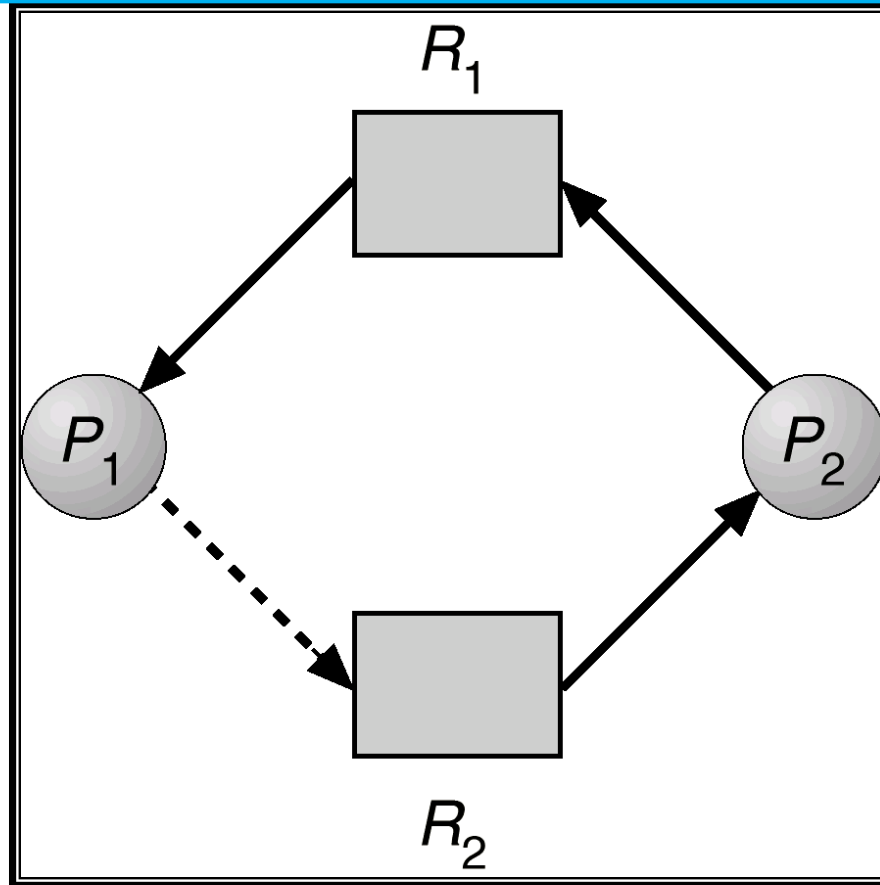
- Αλγόριθμος γράφου κατανομής αγαθών
 - Ακμή διεκδίκησης: διακεκομμένη γραμμή, δείχνει ότι μπορεί να ζητηθεί το αγαθό από τη διεργασία
 - Μετατρέπεται σε ακμή αίτησης όταν το αγαθό ζητηθεί από τη διεργασία
 - Όταν ελευθερωθεί το αγαθό, η ακμή ανάθεσης ξαναγίνεται ακμή διεκδίκησης
 - Τα αγαθά πρέπει να διεκδικούνται εκ των προτέρων στο σύστημα

Αποφυγή αδιεξόδων



Γράφος εκχώρησης αγαθών για αποφυγή αδιεξόδων

Αποφυγή αδιεξόδων



Μη ασφαλής κατάσταση

Αποφυγή αδιεξόδων

- Αλγόριθμος τραπεζίτη (άρνηση εκχώρησης αγαθών)
- Κατάσταση συστήματος: τρέχουσα κατανομή αγαθών στις διεργασίες
- Ασφαλής κατάσταση: υπάρχει τουλάχιστον μια ακολουθία εκχωρήσεων ώστε να μην υπάρξει αδιέξοδο
- Μη ασφαλής κατάσταση: δεν υπάρχει τέτοια ακολουθία

Αποφυγή αδιεξόδων

- Προϋποθέσεις εφαρμογής αλγόριθμου:
 - Μέγιστη απαίτηση σε αγαθά πρέπει να είναι γνωστή εξαρχής
 - Οι διεργασίες πρέπει να είναι ανεξάρτητες (να μην απαιτείται συγχρονισμός)
 - Οι διεργασίες πρέπει να ελευθερώνουν τα αγαθά πριν τερματίσουν

Αποφυγή αδιεξόδων

- Παράδειγμα αλγόριθμου για ένα είδος αγαθού

Έχει Max

A	3	9
B	2	4
C	2	7

Ελεύθερα: 3

Έχει Max

A	3	9
B	4	4
C	2	7

Ελεύθερα: 1

Έχει Max

A	3	9
B	0	-
C	2	7

Ελεύθερα: 5

Έχει Max

A	3	9
B	0	-
C	7	7

Ελεύθερα: 0

Έχει Max

A	3	9
B	0	-
C	0	-

Ελεύθερα: 7

Επίδειξη ότι η πρώτη κατάσταση είναι ασφαλής

Έχει Max

A	3	9
B	2	4
C	2	7

Ελεύθερα: 3

Έχει Max

A	4	9
B	2	4
C	2	7

Ελεύθερα: 2

Έχει Max

A	4	9
B	4	4
C	2	7

Ελεύθερα: 0

Έχει Max

A	4	9
B	0	-
C	2	7

Ελεύθερα: 4

Επίδειξη ότι η δεύτερη κατάσταση είναι μη ασφαλής

Αποφυγή αδιεξόδων

- Παράδειγμα αλγόριθμου για πολλά είδη αγαθών

	R1	R2	R3
P1	3	2	2
P2	6	1	3
P3	3	1	4
P4	4	2	2

Claim matrix C

	R1	R2	R3
P1	1	0	0
P2	6	1	2
P3	2	1	1
P4	0	0	2

Allocation matrix A

	R1	R2	R3
P1	2	2	2
P2	0	0	1
P3	1	0	3
P4	4	2	0

C - A

R1	R2	R3
9	3	6

Resource vector R

R1	R2	R3
0	1	1

Available vector V

(a) Initial state

Αποφυγή αδιεξόδων

- Παράδειγμα αλγόριθμου για πολλά είδη αγαθών

	R1	R2	R3
P1	3	2	2
P2	0	0	0
P3	3	1	4
P4	4	2	2

Claim matrix C

	R1	R2	R3
P1	1	0	0
P2	0	0	0
P3	2	1	1
P4	0	0	2

Allocation matrix A

	R1	R2	R3
P1	2	2	2
P2	0	0	0
P3	1	0	3
P4	4	2	0

C - A

R1	R2	R3
9	3	6

Resource vector R

R1	R2	R3
6	2	3

Available vector V

(b) P2 runs to completion

Αποφυγή αδιεξόδων

- Παράδειγμα αλγόριθμου για πολλά είδη αγαθών

	R1	R2	R3
P1	0	0	0
P2	0	0	0
P3	3	1	4
P4	4	2	2

Claim matrix C

	R1	R2	R3
P1	0	0	0
P2	0	0	0
P3	2	1	1
P4	0	0	2

Allocation matrix A

	R1	R2	R3
P1	0	0	0
P2	0	0	0
P3	1	0	3
P4	4	2	0

C - A

R1	R2	R3
9	3	6

Resource vector R

R1	R2	R3
7	2	3

Available vector V

(c) P1 runs to completion

Αποφυγή αδιεξόδων

- Παράδειγμα αλγόριθμου για πολλά είδη αγαθών

	R1	R2	R3
P1	0	0	0
P2	0	0	0
P3	0	0	0
P4	4	2	2

Claim matrix C

	R1	R2	R3
P1	0	0	0
P2	0	0	0
P3	0	0	0
P4	0	0	2

Allocation matrix A

	R1	R2	R3
P1	0	0	0
P2	0	0	0
P3	0	0	0
P4	4	2	0

C - A

R1	R2	R3
9	3	6

Resource vector R

R1	R2	R3
9	3	4

Available vector V

(d) P3 runs to completion

Αποφυγή αδιεξόδων

- Παράδειγμα αλγόριθμου για πολλά είδη αγαθών

	R1	R2	R3
P1	3	2	2
P2	6	1	3
P3	3	1	4
P4	4	2	2

Claim matrix **C**

	R1	R2	R3
P1	1	0	0
P2	5	1	1
P3	2	1	1
P4	0	0	2

Allocation matrix **A**

	R1	R2	R3
P1	2	2	2
P2	1	0	2
P3	1	0	3
P4	4	2	0

C - A

R1	R2	R3
9	3	6

Resource vector **R**

R1	R2	R3
1	1	2

Available vector **V**

(a) Initial state

	R1	R2	R3
P1	3	2	2
P2	6	1	3
P3	3	1	4
P4	4	2	2

Claim matrix **C**

	R1	R2	R3
P1	2	0	1
P2	5	1	1
P3	2	1	1
P4	0	0	2

Allocation matrix **A**

	R1	R2	R3
P1	1	2	1
P2	1	0	2
P3	1	0	3
P4	4	2	0

C - A

R1	R2	R3
9	3	6

Resource vector **R**

R1	R2	R3
0	1	1

Available vector **V**

(b) P1 requests one unit each of R1 and R3

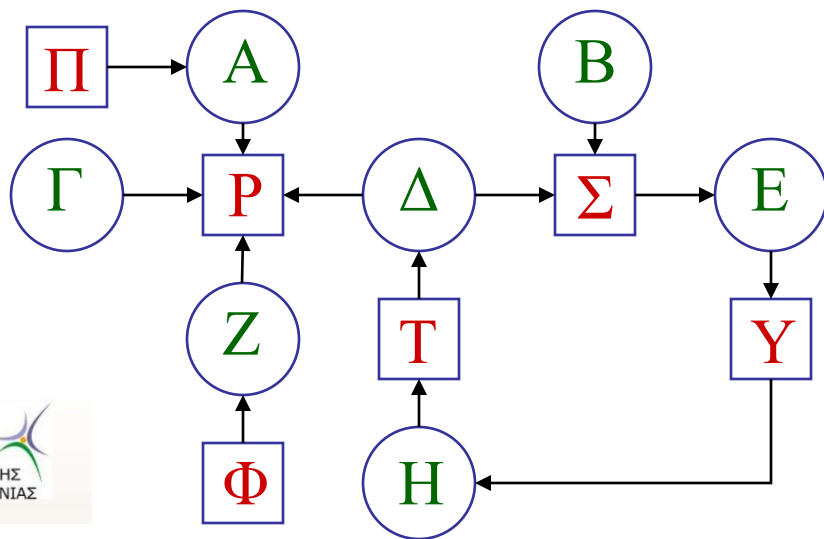
Ανίχνευση αδιεξόδων

- Επιτρέπεται να εμφανιστεί αδιέξοδο
- Πρέπει να υπάρχει αλγόριθμος ανίχνευσης
- Αν ανιχνευτεί αδιέξοδο, πρέπει να ακολουθήσει ανάνηψη από αυτό
- Ανίχνευση κύκλου σε γράφο (μόνο όταν έχω ένα αγαθό από κάθε είδος)

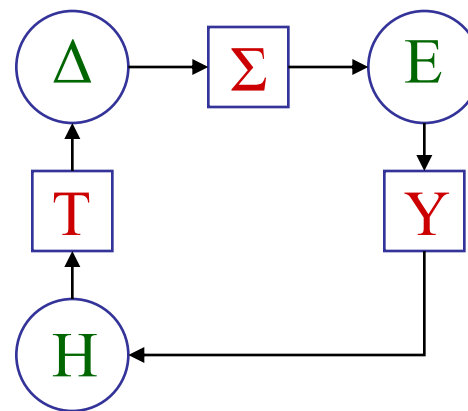
Ανίχνευση αδιεξόδων

- Ανίχνευση κύκλου σε γράφο εκχώρησης αγαθών

1. Για κάθε κόμβο του γράφου εκτέλεσε τα παρακάτω 5 βήματα :
2. Κάνε μία κενή λίστα Λ και θεώρησε όλα τα βέλη ως μη σημειωμένα.
3. Βάλε τον κόμβο στη λίστα. Αν υπάρχει 2 φορές τότε βρέθηκε κύκλος.
4. Αν υπάρχουν εξερχόμενα ασημείωτα βέλη πήγαινε στο 5 αλλιώς στο 6
5. Διάλεξε τυχαία ένα ασημείωτο βέλος, σημείωσέ το, ακολούθησέ το και πήγαινε στο βήμα 3
6. Αν ο κόμβος είναι ο αρχικός τότε το υποδένδρο δεν έχει κύκλους, αλλιώς επέστρεψε στον προηγ. κόμβο και πήγαινε στο βήμα 4.



Κύκλος που ανιχνεύθηκε



Ανίχνευση αδιεξόδων

- Ανίχνευση με χρήση πινάκων:

- Διάνυσμα Y υπαρχόντων πόρων
- Διάνυσμα Θ διαθέσιμων πόρων
- Πίνακας T τρέχουσας κατανομής
- Πίνακας A αιτήσεων

$$\sum_{\kappa=1}^{\nu} T_{\kappa\lambda} + \Theta_{\lambda} = Y_{\lambda}, \forall \lambda \in (1, \mu)$$

Διάνυσμα Υπαρχόντων Πόρων

$$Y = (Y_1, Y_2, Y_3, \dots, Y_{\mu})$$

Πίνακας Τρέχουσας Κατανομής

T_{11}	T_{12}	T_{13}	...	$T_{1\mu}$
T_{21}	T_{22}	T_{23}	...	$T_{2\mu}$
...
$T_{\nu 1}$	$T_{\nu 2}$	$T_{\nu 3}$...	$T_{\nu \mu}$

Διάνυσμα Διαθέσιμων Πόρων

$$\Theta = (\Theta_1, \Theta_2, \Theta_3, \dots, \Theta_{\mu})$$

Πίνακας Αιτήσεων

A_{11}	A_{12}	A_{13}	...	$A_{1\mu}$
A_{21}	A_{22}	A_{23}	...	$A_{2\mu}$
...
$A_{\nu 1}$	$A_{\nu 2}$	$A_{\nu 3}$...	$A_{\nu \mu}$

Ανίχνευση αδιεξόδων

Ορίζοντας την σχέση διανυσμάτων $A \leq B$ όταν $A_k \leq B_k$, $1 \leq k \leq \mu$, ο αλγόριθμος ανίχνευσης αδιεξόδων είναι:

1. Αρχικά όλες οι διεργασίες $\Delta_1, \dots, \Delta_n$ θεωρούνται μη σημειωμένες.
2. Ψάξε για μία ασημείωτη διεργασία Δ_k , για την οποία η k γραμμή του Πίνακα Αιτήσεων A να είναι μικρότερη ή ίση από το Διάνυσμα Διαθέσιμων Πόρων Θ .
3. Αν βρεθεί, τότε πρόσθεσε την k γραμμή του Πίνακα Τρέχουσας Κατανομής T , στο Διάνυσμα Διαθέσιμων Πόρων Θ , σημείωσε την διεργασία και πήγαινε στο βήμα 2.
4. Αν δεν υπάρχει τέτοια διεργασία ο αλγόριθμος τερματίζεται

Στο τέλος του αλγορίθμου όλες οι μη σημειωμένες διεργασίες, αν υπάρχουν, βρίσκονται σε αδιέξοδο. Παράδειγμα για 3 διεργασίες:

	HD	CD	USB	PRN		0 0 1 0		2 0 0 1
$Y =$	(4	2	3	1)	$T =$	2 0 0 1	$A =$	1 0 1 0
$\Theta =$	(2	1	0	0)		0 1 2 0		2 1 0 0

Ανάληψη από αδιέξοδο

- Αν ανιχνευτεί αδιέξοδο, τότε μπορούμε:
 - να τερματίσουμε όλες τις εμπλεκόμενες διεργασίες
 - να τερματίσουμε μία-μία τις διεργασίες μέχρι να αρθεί το αδιέξοδο
- Ποια διεργασία θα επιλέξουμε;
 - Μικρότερη προτεραιότητα
 - Αυτή που έτρεξε λιγότερο
 - Αυτή που κρατά τα λιγότερα αγαθά
 - Τις ομαδικής επεξεργασίας σε βάρος των διαδραστικών
 - Όσο γίνεται λιγότερες

Ανάληψη από αδιέξοδο

- Ανάληψη μέσω προεκτόπισης και οπισθοδρόμησης
 - Αφαίρεσε σταδιακά αγαθά από τις διεργασίες μέχρι να μην υπάρχει πλέον αδιέξοδο
 - Κατέγραφε σημεία ελέγχου (σημεία πριν γίνουν νέες αναθέσεις αγαθών) ώστε να μπορούν να ξαναεκτελεστούν από εκεί οι διεργασίες που εμπλέκονται

Τέλος Ενότητας

