



# Λειτουργικά Συστήματα

Ενότητα 5β: Διαχείριση Κύριας Μνήμης – Διαχείριση  
Δευτερεύουσας Μνήμης

Αθηνά Βακάλη  
Τμήμα Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



# 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 Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» 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.



# Δευτερεύουσα μνήμη: Βελτιώνει το βαθμό χρήσης της Κύριας μνήμης

Τα τμήματα μιας διεργασίας **δεν είναι απαραίτητο** να βρίσκονται όλα ταυτόχρονα **στην κύρια μνήμη**. Η παρουσία τους ή όχι εξαρτάται από τις συνθήκες εκτέλεσης της διεργασίας τη συγκεκριμένη στιγμή.

Τα τμήματα που **δεν είναι απαραίτητο** να υπάρχουν βρίσκονται στη δευτερεύουσα μνήμη (σκληρός δίσκος) και χρειάζεται να μεταφερθούν στη μνήμη όταν απαιτηθούν.

Βελτιώνεται ο βαθμός χρήσης της κύριας μνήμης.

- Δε βρίσκονται στη μνήμη τμήματα που χρησιμοποιούνται σπάνια.
- Μπορούν να εξυπηρετηθούν περισσότερες διεργασίες.

Η διαχείριση πρέπει να γίνει με προσοχή για να αποφευχθεί σημαντική μείωση της απόδοσης.



# Σκοπός της διαχείρισης ιδεατής μνήμης

Η κατανόηση των βασικών τεχνικών της εικονικής μνήμης.

- Μέθοδοι προσκόμισης των ζητούμενων σελίδων.
- Μέθοδοι αντικατάστασης σελίδων.
- Εκχώρηση πλαισίων στις διεργασίες.

Η αναφορά σε περιορισμούς και στόχους.

- Τοπικότητα της αναφοράς.
- Πρόβλεψη των σελίδων που θα ζητηθούν μελλοντικά.

Ο υπολογισμός της επίδρασης που έχει ο βαθμός απόδοσης της ιδεατής μνήμης στην συνολική απόδοση του συστήματος.



# Συστήματα χωρίς ιδεατή μνήμη

## Περιορισμοί

Η μνήμη δε χρησιμοποιείται πλήρως.

Χαμηλός βαθμός πολυπρογραμματισμού.

- Δεν υπάρχει αρκετή φυσική μνήμη για την εξυπηρέτηση όλων των διεργασιών.

Μέγεθος προγραμμάτων.

- Το μέγεθος των προγραμμάτων (process images) περιορίζεται από τη διαθέσιμη φυσική μνήμη.

Επίπεδο αφαιρετικότητας.

- Ο προγραμματιστής πρέπει να λαμβάνει υπόψη του τις λεπτομέρειες του υλικού, π.χ. το μέγεθος της φυσικής μνήμης.



# Λογισμικό του Λειτουργικού Συστήματος για διαχείριση μνήμης

- Η σχεδίαση του συστήματος διαχείρισης μνήμης του Λειτουργικού Συστήματος βασίζεται σε τρεις θεμελιώδεις περιοχές επιλογών:
  - Χρήση (ή όχι) τεχνικών ιδεατής μνήμης.
  - Χρήση σελιδοποίησης ή κατάτμησης ή συνδυασμού και των δύο.
  - Χρήση αλγορίθμων για θέματα διαχείρισης μνήμης.
- Οι επιλογές για τα 2 πρώτα εξαρτώνται από το υλικό.
- Η σελιδοποίηση αφορά τα περισσότερα θέματα διαχείρισης μνήμης.



# Σχεδίαση Λογισμικού

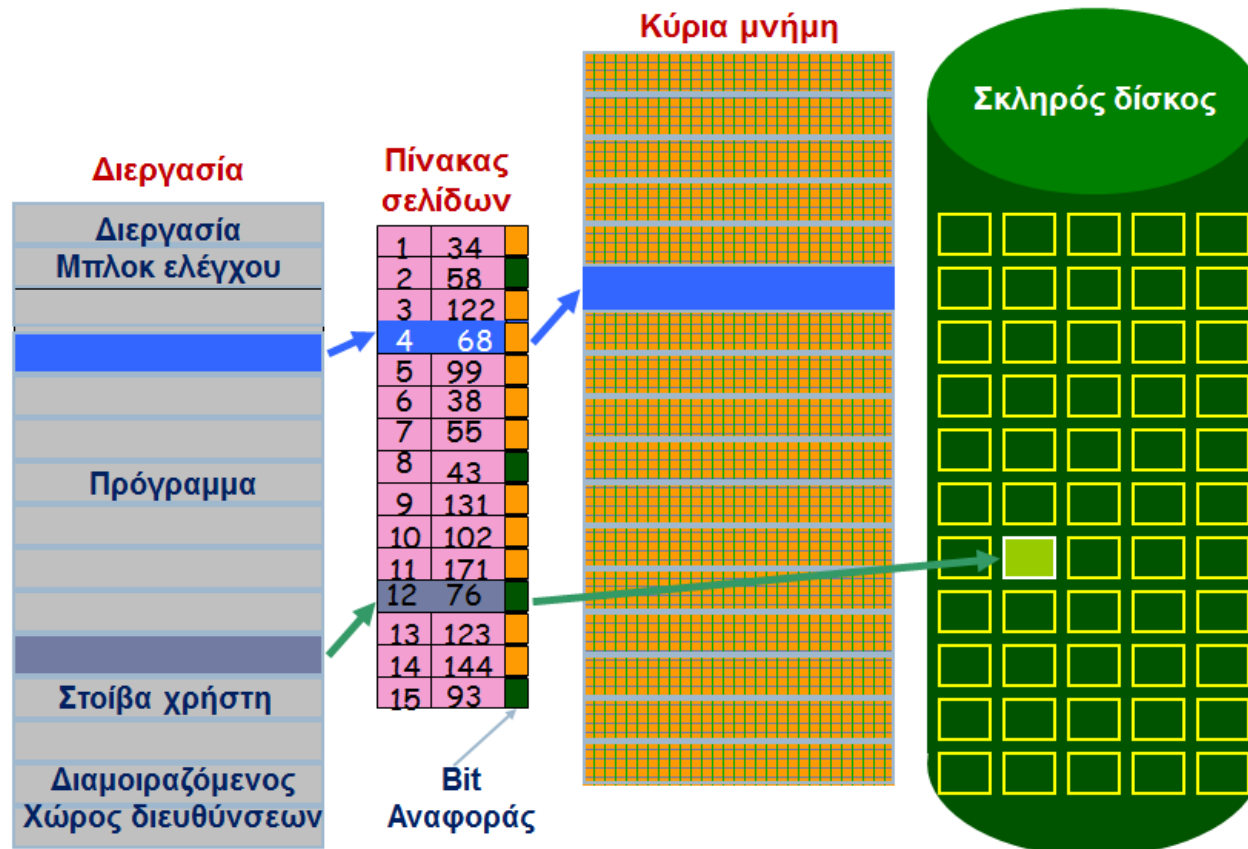
Βασικά θέματα σχεδίασης του λογισμικού του Λειτουργικού Συστήματος.

- Περιορισμός του ρυθμού εμφάνισης των σφαλμάτων σελίδας (επιπλέον φόρτος εργασίας).
- Δρομολόγηση διεργασιών και εναλλαγή διεργασιών.
- Δεν υπάρχουν βέλτιστες πολιτικές διαχείρισης.
- Η επίδοση κάθε μεθόδου εξαρτάται από το μέγεθος της κύριας μνήμης, την ταχύτητα κύριας και δευτερεύουσας μνήμης, το μέγεθος και το πλήθος των διεργασιών που ανταγωνίζονται για πόρους **και τη συμπεριφορά της διεργασίας καθώς εκτελείται.**
- Στα μεγάλα συστήματα υπάρχουν εργαλεία καταγραφής και ελέγχου.





# Εικονική μνήμη (1/2)

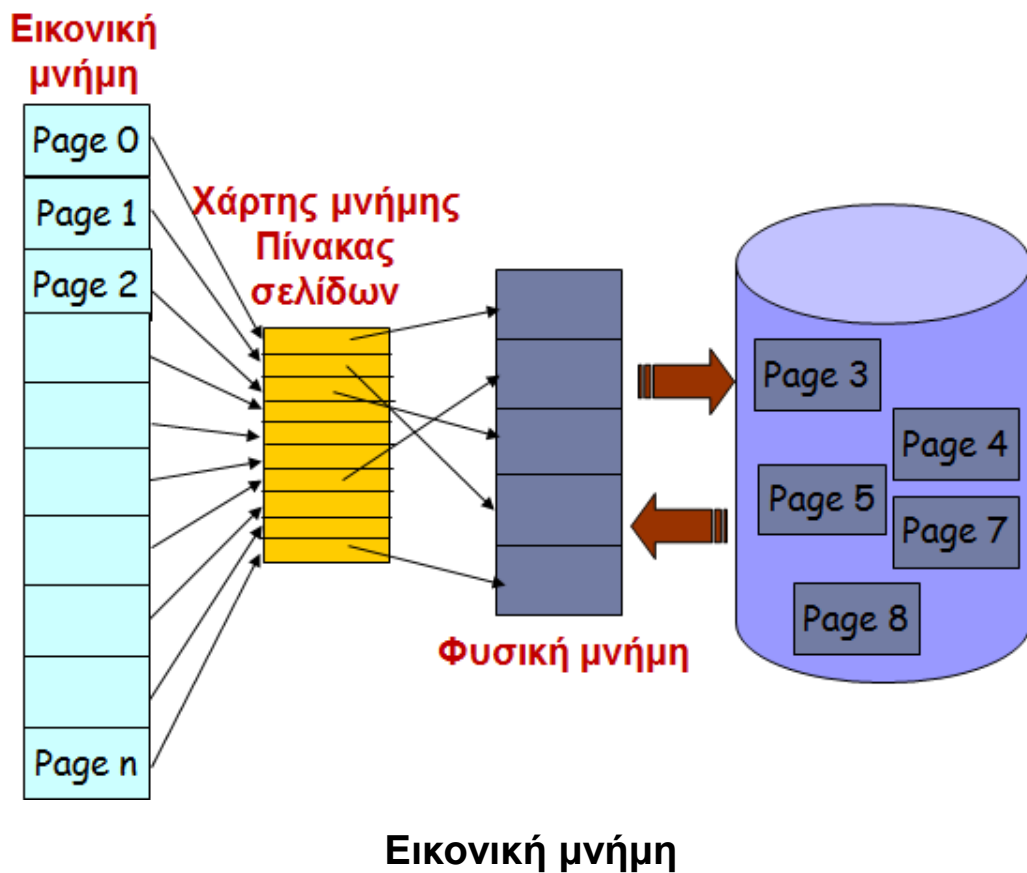


## Εικονική μνήμη

Κύρια Μνήμη – Δευτερεύουσα Μνήμη

Τμήμα Πληροφορικής

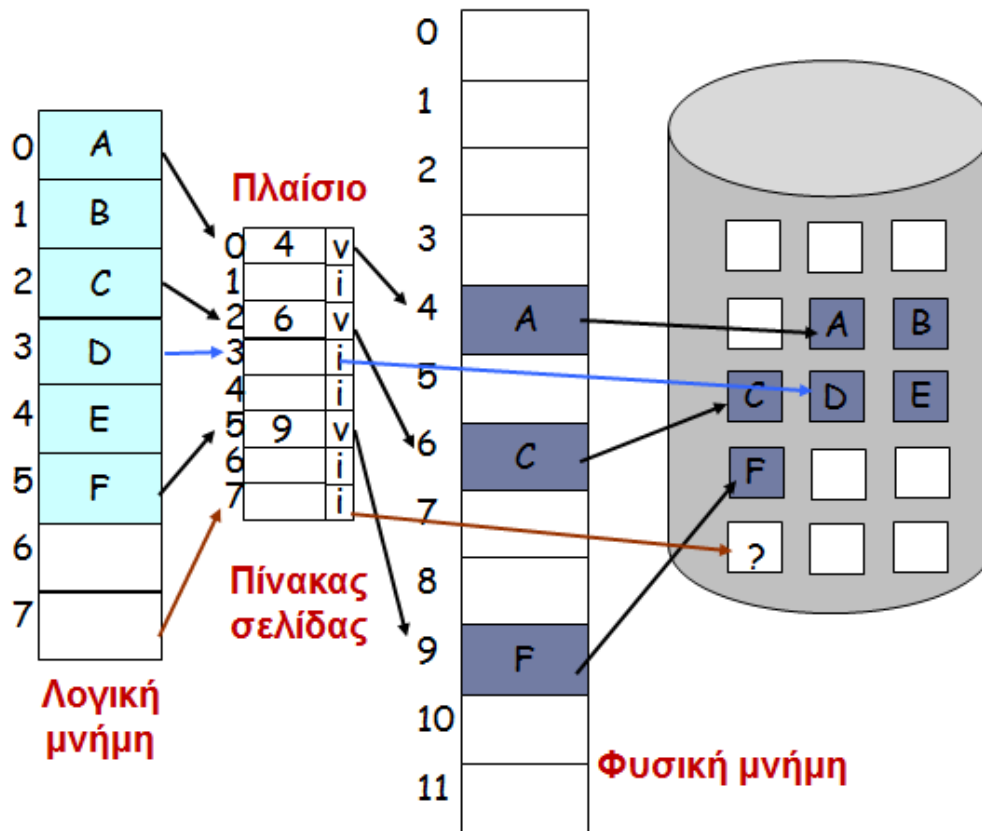
# Εικονική μνήμη (2/2)



- Αναφορά σελίδας.
- Έλεγχος χάρτη μνήμης.
- Αν υπάρχει αντίστοιχη καταχώρηση στον πίνακα σελίδων για πλαίσιο φυσικής μνήμης τότε γίνεται αναφορά σ' αυτό.
- Διαφορετικά, η σελίδα δεν υπάρχει στη φυσική μνήμη και πρέπει να μεταφερθεί από το δίσκο.
- Αν η φυσική μνήμη είναι πλήρης:
  - Εύρεση σελίδας προς αντικατάσταση.
  - Εναλλαγή στον δίσκο.



# Διαχείριση πίνακα σελίδων εικονικής μνήμης



- Αν το bit πλαισίου είναι έγκυρο (v - valid), το πλαίσιο βρίσκεται στη μνήμη.
- Διαφορετικά, συμβαίνει σφάλμα σελίδας:
  - Η αντίστοιχη σελίδα φορτώνεται από το δίσκο.
  - Εκχωρείται μια νέα περιοχή μνήμης.

## Διαχείριση πίνακα σελίδων εικονικής μνήμης



# Διατήρηση στοιχείων διεργασιών στην Κύρια μνήμη

Στην πραγματικότητα δεν είναι απαραίτητο να διατηρείται η πλήρης εικόνα μιας διεργασίας συνεχώς στην κύρια μνήμη. Διατηρούνται :

- Ο πρόσφατα χρησιμοποιούμενος κώδικας.
- Οι πρόσφατα χρησιμοποιούμενες δομές δεδομένων.
- Δεδομένα του συστήματος (heap, stack).

Ορισμένα τμήματα της εικόνας της διεργασίας μπορούν να βρίσκονται στη δευτερεύουσα μνήμη.

- Πρέπει να γίνει εναλλαγή όταν ζητηθεί ένα τμήμα που δε βρίσκεται στην κύρια μνήμη.



# Χειρισμός διακοπής

## Page fault

Όταν μια διεύθυνση που απαιτείται δε βρίσκεται στην κεντρική μνήμη δημιουργείται μια διακοπή (**interrupt**).

Το Λειτουργικό Σύστημα θέτει τη διεργασία σε κατάσταση αναστολής (**blocking**).

Το τμήμα της διεργασίας που περιέχει τη λογική διεύθυνση μεταφέρεται στην κεντρική μνήμη.

- Το Λειτουργικό Σύστημα θέτει μια αίτηση ανάγνωσης δίσκου (**disk I/O**).
- Μια άλλη διεργασία δρομολογείται και εκτελείται, ενώ πραγματοποιείται η διεργασία disk I/O.
- Μια διακοπή προκαλείται όταν η διαδικασία disk I/O ολοκληρωθεί που αναγκάζει το Λειτουργικό Σύστημα να μεταφέρει τη διεργασία σε κατάσταση Ready.



# Απαιτήσεις υλικού για την υποστήριξη της ιδεατής μνήμης

## Απαιτήσεις υλικού

Το Hardware πρέπει να υποστηρίζει σελιδοποίηση ή/και κατάτμηση και να διαθέτει μονάδα διαχείρισης μνήμης (MMU).

## Ελάχιστες απαιτήσεις

- Υποστήριξη μετατροπής διευθύνσεων (σελιδοποίηση & κατάτμηση).
- Οι πίνακες σελίδων πρέπει να περιέχουν επιπρόσθετες πληροφορίες (για παράδειγμα present bit, modify bit).



# Απαιτήσεις λογισμικού για την υποστήριξη της ιδεατής μνήμης (1/2)

Το Λειτουργικό Σύστημα πρέπει να:

- Έχει την ικανότητα της **διαχείρισης των μετακινήσεων** των σελίδων ή/και των τμημάτων μεταξύ δευτερεύουσας και κεντρικής μνήμης.
- Υποστηρίζει **σελιδοποίηση** και **κατάτμηση**.
- Μπορεί να **εναλλάσσει (swap)** (μέσω φόρτωσης και απομάκρυνσης) τμήματα διεργασιών μεταξύ κύριας μνήμης και σκληρού δίσκου.



# Απαιτήσεις λογισμικού για την υποστήριξη της ιδεατής μνήμης (2/2)

Το Λειτουργικό Σύστημα πρέπει να:

- Διαθέτει **αλγόριθμο προσκόμισης** (**fetch algorithm**)
  - Ο αλγόριθμος προσκόμισης αποφασίζει ποια τμήματα διεργασιών θα πρέπει να μεταφερθούν στην κύρια μνήμη.
  - Η απαίτηση σελιδοποίησης (**demand paging**) είναι η πιο συχνά χρησιμοποιούμενη.
- Διαθέτει **αλγόριθμο αντικατάστασης** που θα αποφασίζει ποια τμήματα διεργασιών θα εναλλαγούν όταν μια ποσότητα μνήμης πρέπει να απελευθερωθεί.





# Δομές δεδομένων (1/2)

- Σελίδες ή τμήματα.
- Πλαίσια σελίδων.
- Πίνακες σελίδων.
- Λίστα των ελεύθερων πλαισίων σελίδων.
  - Περιέχει όλα τα πλαίσια που δε χρησιμοποιούνται.
- Πίνακες αλγορίθμου αντικατάστασης.
  - Περιέχουν δεδομένα που αφορούν σελίδες και πλαίσια.
  - Χρησιμοποιούνται για να αποφασιστεί ποιες σελίδες θα αντικατασταθούν.



# Δομές δεδομένων (2/2)

- Περιοχή εναλλαγής.
  - Περιοχή της δευτερεύουσας μνήμης για σελίδες που δε βρίσκονται στην κύρια μνήμη.
  - Μια πλήρης εικόνα κάθε διεργασίας διατηρείται στην περιοχή αυτή.



# Πλεονεκτήματα και Προβλήματα (1/2)

## Πλεονεκτήματα

- Μεγάλος εικονικός χώρος διευθύνσεων.
  - Οι διεργασίες **μπορούν να είναι μεγαλύτερες** από τη φυσική μνήμη.
- Καλύτερη χρήση της μνήμης.
  - **Μόνον τα τμήματα της διεργασίας που χρησιμοποιούνται** πραγματικά διατηρούνται στην κύρια μνήμη.
- **Λιγότερο I/O για φόρτωση** ή εναλλαγή της διεργασίας.
  - Μεταφέρονται **μόνον τα τμήματα της διεργασίας που χρειάζονται**.



# Πλεονεκτήματα και Προβλήματα (2/2)

## Προβλήματα

- Σύνθετη (**πολύπλοκη**) υλοποίηση.
  - Επιπλέον hardware, συστατικά και δομές δεδομένων.
- Απώλεια στην **απόδοση**.
  - Χρονική επιβάρυνση εξ αιτίας της διαχείρισης.
  - Καθυστέρηση κατά τη μεταφορά σελίδων.



# Σφάλματα σελίδα (Page fault)

Όταν μια αναφορά στη μνήμη οδηγεί σε μια σελίδα που δε βρίσκεται στην κύρια μνήμη δημιουργείται σφάλμα σελίδας ([interrupt/trap](#)).

- Μπορεί να προκληθεί από κάθε είδους πρόσβαση στη μνήμη (instructions, user data, system data).
- Ο πίνακας σελίδων πρέπει να διατηρεί επιπρόσθετες πληροφορίες για να βρεθεί αν η σελίδα βρίσκεται στην κύρια μνήμη.
  - Present bit.
  - Valid/invalid bit.

Η σελίδα χρειάζεται να φορτωθεί ώστε να συνεχιστεί η εκτέλεση της διεργασίας.



# Διαχείριση σφαλμάτων σελίδας

- Δημιουργείται ένα σφάλμα σελίδας αν το present bit δεν έχει τιμή.
- Εκχωρείται ένα πλαίσιο σελίδας.
  - Αν δεν υπάρχουν ελεύθερα πλαίσια πρέπει να κληθεί ο αλγόριθμος αντικατάστασης σελίδων.
- Μια σελίδα διαβάζεται και καταλαμβάνει το πλαίσιο σελίδας.
- Ο πίνακας σελίδων ενημερώνεται.
- Η εντολή επανεκκινεί.



# Χρόνοι σφαλμάτων σελίδας

| Βήμα                | Δράση   | Χρόνος                               |
|---------------------|---|--------------------------------------|
| Interrupt handling  | A few hundred to a few thousand instructions. | $2000 * 10\text{ns} = 20\mu\text{s}$ |
| Frame allocation    | A few hundred to a few thousand instructions. | $2000 * 10\text{ns} = 20\mu\text{s}$ |
| Page replacement    | Some instructions, write a page from disk.    | $\approx 20\text{ms}$                |
| Page loading        | Some instructions, read a page from disk.     | $\approx 20\text{ms}$                |
| Page table update   | A few hundred instructions.                   | $500 * 10\text{ns} = 5\mu\text{s}$   |
| Instruction restart | A few instructions.                           | $30 * 10\text{ns} = 0.3\mu\text{s}$  |
| Συνολικό χρόνο      | Dominated by Input/Output operations.         | $\approx 20 \dots 40\text{ms}$       |

**Υποθέσεις:** 100 MHz CPU clock cycle, 20ms μέσος χρόνος προσπέλασης και μεταφοράς ανά σελίδα.



# Ρυθμός σφάλματος σελίδας (page fault rate)

- Μια εντολή που κανονικά χρειάζεται μερικές δεκάδες nanoseconds θα χρειαστεί δεκάδες milliseconds αν συμβεί σφάλμα σελίδας.
  - Ο συντελεστής είναι 100.000 μεγαλύτερος.
  - Η καθυστέρηση είναι υπερβολική (ίσως και αφόρητη).
- Η συχνότητα εμφάνισης των σφαλμάτων σελίδας είναι πολύ σημαντική για την απόδοση του συστήματος.
- Ο ρυθμός σφαλμάτων σελίδας πρέπει να παραμένει χαμηλός.





# Σφάλματα σελίδας και πραγματικός χρόνος προσπέλασης

- Πραγματικός χρόνος προσπέλασης (Effective Access Time - EAT).
- $p$  = ρυθμός σφαλμάτων σελίδας (page fault rate).

$$\begin{aligned} \text{EAT} \\ = \\ p * \text{προσπέλαση με σφάλμα σελίδας} \\ + \\ (1-p) * \text{προσπέλαση χωρίς σφάλμα σελίδας} \end{aligned}$$



# Σφάλματα σελίδας και ομαδοποίηση

## Άσκηση

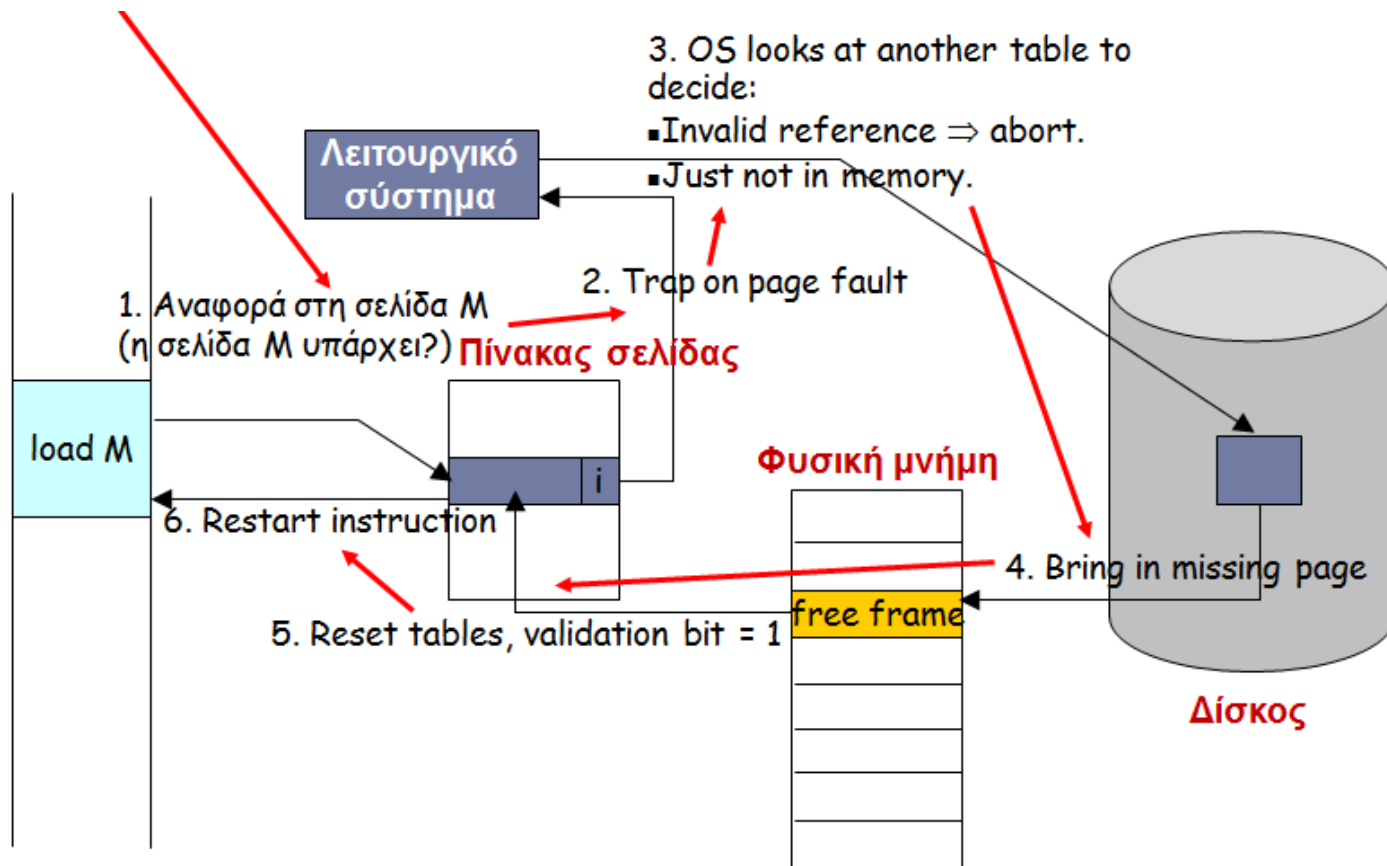
Ποια είναι η απώλεια απόδοσης για ρυθμό σφαλμάτων σελίδας 1 σε 100,000;

- Χρόνος προσπέλασης χωρίς σφάλμα σελίδας: 100 ns.
  - Χρόνος προσπέλασης μνήμης συμπεριλαμβάνοντας τον πίνακα σελίδων και το TLB.
- Χρόνος προσπέλασης με σφάλμα σελίδας και με αντικατάσταση σελίδας για το 50% των σφαλμάτων σελίδας :
  - (ΔΕΔΟΜΕΝΑ :20 ms ο χρόνος φόρτωσης μιας σελίδας σε ελεύθερο πλαίσιο και 20 ms ο χρόνος αντικατάστασης σελίδας λόγω μη ύπαρξης ελεύθερου πλαισίου).
  - $0.5 * 20 \text{ ms} + 0.5 * 40 \text{ ms} = 30 \text{ ms}$ .

$$\begin{aligned} \text{EAT} &= 0.00001 * 30 \text{ ms} + (1 - 0.00001) * 100 \text{ ns} \\ &= 300 \text{ ns} + 99.999999 \text{ ns} \approx 400 \text{ ns} : \text{Απώλεια } 300\% \end{aligned}$$

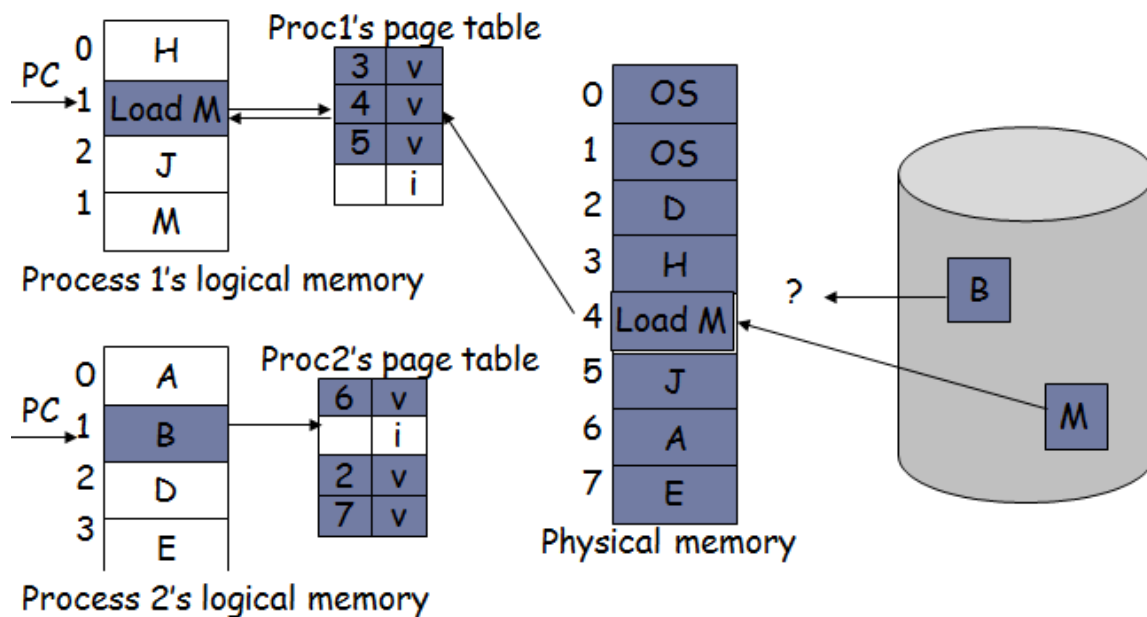


# Χειρισμός Page fault



## Χειρισμός Page fault

# Εύρεση του πλαισίου μνήμης που θα μεταφερθεί στο Swap



Έρευνα πλαισίου μνήμης

Δεν υπάρχουν ελεύθερα πλαίσια.

Εύρεση μιας σελίδας προς αντικατάσταση.

- Αν φύγει η M θα προκληθεί σφάλμα σελίδας.
- Αν φύγει η H πιθανόν να εξυπηρετεί.
- Ο αλγόριθμος αντικατάστασης πρέπει να ελαχιστοποιεί τα σφάλματα σελίδας.



# Τοπικότητες της αναφοράς

Δείχνει ότι η επόμενη προσπέλαση στη μνήμη θα γίνει σε περιοχή της μνήμης γειτονική με την τρέχουσα αναφορά μνήμης.

- **Χωρική**
  - Διαδοχικές προσπελάσεις στη μνήμη θα γίνονται σε γειτονικές περιοχές.
- **Χρονική**
  - Επαναληπτική προσπέλαση στις ίδιες θέσεις μνήμης.

Η τοπικότητα της αναφοράς είναι η βασική αιτία για το χαμηλό πλήθος σφαλμάτων σελίδας.



# Λόγοι που εξηγούν την τοπικότητα της αναφοράς

## Εκτέλεση εντολών.

- Σειριακή εκτέλεση εντολών σε προγράμματα.
  - Εξαιρούνται οι διακλαδώσεις και οι κλήσεις συναρτήσεων (εκτός των συναρτήσεων inline).

## Πρόσβαση σε δεδομένα.

- Πολλές δομές δεδομένων υλοποιούνται σειριακά.
  - List, array, tree.

Το στυλ προγραμματισμού και ο κώδικας που παράγεται από τους compilers.



# Πολιτικές του Λειτουργικού Συστήματος για τη διαχείριση μνήμης

## Παραλαβής ή προσκόμισης (Fetch policy)

- Καθορίζει πότε μια σελίδα πρέπει να έλθει στην κύρια μνήμη.

## Τοποθέτησης (Placement policy)

- Καθορίζει σε ποιο σημείο της μνήμης θα αποθηκευθεί ένα τμήμα της διεργασίας.

## Αντικατάστασης (Replacement policy)

- Καθορίζει ποια σελίδα θα αντικατασταθεί.



# Πολιτική προσκόμισης (Fetch policy)

Καθορίζει **πότε** μια σελίδα πρέπει να μεταφερθεί στη μνήμη.

Απαίτηση σελιδοποίησης (**Demand paging**).

- Μεταφέρει σελίδες στην κύρια μνήμη μόνον όταν γίνει μια αναφορά σε μια διεύθυνση της σελίδας και αυτή δεν υπάρχει στη μνήμη.
  - Δημιουργούνται πολλά σφάλματα σελίδας (page faults) όταν η διεργασία ξεκινά για πρώτη φορά.

Προ-σελιδοποίηση (**pre-paging**).

- Μεταφέρει περισσότερες σελίδες από αυτές που χρειάζονται.
  - Είναι περισσότερο αποτελεσματική η μεταφορά σελίδων που βρίσκονται συνεχόμενες στο δίσκο, αλλά δεν είναι πάντοτε εφικτή, εξαρτώμενη από τις πολιτικές αποθήκευσης στη δευτερεύουσα μνήμη.





# Πολιτική τοποθέτησης (Placement policy)

- Προσπαθεί να εξακριβώσει μια καλή θέση για το τμήμα του προγράμματος που θα μεταφερθεί στην κύρια μνήμη.
- Δεν αποτελεί πρόβλημα στη σελιδοποίηση διότι κάθε σελίδα ταιριάζει σε οποιοδήποτε πλαίσιο.
- Στην απλή κατάτμηση είναι ουσιαστικά μια παραλλαγή του γενικού προβλήματος παραχώρησης μνήμης.
  - Εύρεση μιας κενής περιοχής που ταιριάζει για τη μεταφορά του τμήματος.



# Πολιτική αντικατάστασης (Replacement policy)

Ασχολείται με την επιλογή μιας σελίδας στη μνήμη, η οποία θα αντικατασταθεί μόλις θα πρέπει να φορτωθεί μια νέα σελίδα.

Το θέμα σχετίζεται και με πλήθος άλλων, όπως:

- Το πλήθος των πλαισίων που ανατίθενται σε κάθε ενεργή διεργασία.
- Το σύνολο των υποψηφίων προς αντικατάσταση σελίδων.
  - Διαχείριση του παραμένοντος συνόλου.

**Ποια σελίδα θα αντικατασταθεί;**



# Ποια σελίδα θα αντικατασταθεί

- Η σελίδα που θα μετακινηθεί θα είναι εκείνη η σελίδα που θα έχει την ελάχιστη πιθανότητα να χρησιμοποιηθεί στο άμεσο μέλλον.
- Οι περισσότερες πολιτικές προβλέπουν τη μελλοντική συμπεριφορά βασιζόμενες στην προηγούμενη συμπεριφορά.
- Η αρχή της τοπικότητας χρησιμοποιήθηκε σε πολλές πολιτικές.
- Η πολυπλοκότητα της μεθόδου επιφέρει αντίστοιχη επιβάρυνση στο υλικό και το λογισμικό που θα την υλοποιήσει.



# Βασικοί αλγόριθμοι αντικατάστασης σελίδων

- First-in, First-out (**FIFO**).
- Βέλτιστη πολιτική (optimal).
- Λιγότερο πρόσφατα χρησιμοποιούμενη σελίδα - Least Recently Used (**LRU**).



# Αποτίμηση των αλγορίθμων

- Διατήρηση του πλήθους των σφαλμάτων σελίδων όσο γίνεται χαμηλά.
- Η απόδοση των αλγορίθμων αντικατάστασης κρίνεται από τα σφάλματα που παράγονται σε σχέση με ένα αλφαριθμητικό αναφοράς (**reference string**) που είναι μια ακολουθία αναφοράς σελίδων.
  - Παράγεται από την καταγραφή διευθύνσεων του συστήματος στις οποίες γίνεται αναφορά.
- Εξαρτάται από το πλήθος των διαθέσιμων πλαισίων.
  - Περισσότερα πλαίσια μειώνουν το πλήθος σφαλμάτων σελίδας.



# First in First out (FIFO)

- Αντιμετωπίζει τα πλαίσια σελίδων που ανατίθενται σε μια διεργασία, ως ένα κυκλικό ενδιάμεσο χώρο αποθήκευσης (**circular buffer**).
- Οι σελίδες αντικαθίστανται με κυκλική σειρά (**round-robin**).
- Αντικαθίσταται η σελίδα που παρέμεινε στη μνήμη για το μεγαλύτερο χρονικό διάστημα.
- Η σελίδα που μεταφέρθηκε πρώτη είναι η πρώτη που αντικαθίσταται.
  - Πρέπει να αποθηκεύεται η χρονική στιγμή φόρτωσης της σελίδας στη μνήμη.
- Είναι πολύ απλή, αλλά όχι πολύ καλή.
  - Η παλαιότερη σελίδα (που στο μεταξύ έχει αντικατασταθεί) είναι πιθανόν να απαιτηθεί εκ νέου πολύ σύντομα.



# Παράδειγμα (1/4)

Αλφαριθμητικό αναφοράς: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5.

3 Πλαίσια/Διεργασία  
(Frames/Process)

|   |   |   |
|---|---|---|
| 1 | 4 | 5 |
| 2 | 1 | 3 |
| 3 | 2 | 4 |

9 Σφάλματα σελίδας  
(Page faults)

4 Πλαίσια/Διεργασία  
(Frames/Process)

|   |   |   |
|---|---|---|
| 1 | 5 | 4 |
| 2 | 1 | 5 |
| 3 | 2 |   |
| 4 | 3 |   |

10 Σφάλματα σελίδας  
(Page faults)



# Παράδειγμα (2/4)

Αν η μνήμη αποτελείται από 3 πλαίσια και η διεργασία έχει 5 σελίδες, με την ακόλουθη χρονική σειρά αναφοράς:

**/2/3/2/1/5/2/4/5/3/2/5/2/**

Τότε θα έχουμε τα παρακάτω σφάλματα σελίδας (σημειώνονται με \*):

**/2\*/3\*/2/1\*/5\*/2\*/4\*/5/3\*/2/5\*/2\*/**





# Παράδειγμα (3/4)

Έστω ότι έχουμε:

- Επτά σελίδες
- Τέσσερα πλαίσια

## Αλφαριθμητικό αναφοράς

|  |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
|--|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|  | <b>4</b> | <b>3</b> | <b>1</b> | <b>5</b> | <b>1</b> | <b>2</b> | <b>3</b> | <b>6</b> | <b>7</b> | <b>4</b> | <b>2</b> | <b>5</b> | <b>6</b> | <b>1</b> | <b>3</b> | <b>4</b> | <b>7</b> |
|  | 4        | 4        | 4        | 4        | 4        | 2        | 2        | 2        | 2        | 2        | 2        | 5        | 5        | 5        | 5        | 5        | 5        |
|  |          | 3        | 3        | 3        | 3        | 3        | 3        | 6        | 6        | 6        | 6        | 6        | 6        | 1        | 1        |          |          |
|  |          |          | 1        | 1        | 1        | 1        | 1        | 1        | 7        | 7        | 7        | 7        | 7        | 7        | 3        | 3        | 3        |
|  |          |          |          | 5        | 5        | 5        | 5        | 5        | 5        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 7        |
|  | <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> |          | <b>F</b> |          | <b>F</b> | <b>F</b> | <b>F</b> |          | <b>F</b> |          | <b>F</b> | <b>F</b> |          | <b>F</b> |

**Σφάλματα σελίδας: 12**



# Παράδειγμα (4/4)

Έστω ότι έχουμε:

- Επτά σελίδες
- Πέντε πλαίσια

## Αλφαριθμητικό αναφοράς

|  |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
|--|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|  | <b>4</b> | <b>3</b> | <b>1</b> | <b>5</b> | <b>1</b> | <b>2</b> | <b>3</b> | <b>6</b> | <b>7</b> | <b>4</b> | <b>2</b> | <b>5</b> | <b>6</b> | <b>1</b> | <b>3</b> | <b>4</b> | <b>7</b> |
|  | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 6        | 6        | 6        | 6        | 6        | 6        | 6        | 6        | 6        | 6        |
|  |          | 3        | 3        | 3        | 3        | 3        | 3        | 3        | 7        | 7        | 7        | 7        | 7        | 7        | 7        | 7        | 7        |
|  |          |          | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        |
|  |          |          |          | 5        | 5        | 5        | 5        | 5        | 5        | 5        | 5        | 5        | 5        | 1        | 1        | 1        | 1        |
|  |          |          |          |          |          | 2        | 2        | 2        | 2        | 2        | 2        | 2        | 2        | 2        | 3        | 3        | 3        |
|  | <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> |          | <b>F</b> |          | <b>F</b> | <b>F</b> | <b>F</b> |          |          |          | <b>F</b> | <b>F</b> |          |          |

**Σφάλματα σελίδας: 10**



# Πολιτικές αντικατάστασης

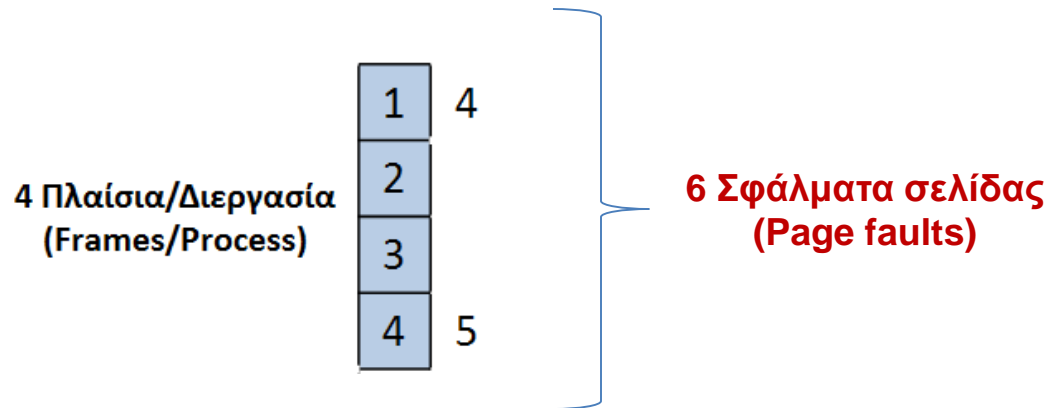
## Βέλτιστη πολιτική

- Επιλέγεται για αντικατάσταση η σελίδα στην οποία θα γίνει αναφορά το αργότερο δυνατό στο μέλλον.
- Είναι αδύνατο να υπάρξει πλήρης γνώση των μελλοντικών συμβάντων άρα ως μέθοδος είναι αδύνατον να υλοποιηθεί και επομένως δε χρησιμοποιείται σε πραγματικά συστήματα.
- Χρησιμοποιείται ως benchmark για την αποτίμηση της απόδοσης άλλων αλγορίθμων αντικατάστασης.



# Παράδειγμα (1/9)

Αλφαριθμητικό αναφοράς: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5.



# Παραδείγματα (2/9)

Αν η μνήμη αποτελείται από 3 πλαίσια και η διεργασία έχει 5 σελίδες, με την ακόλουθη χρονική σειρά αναφοράς:

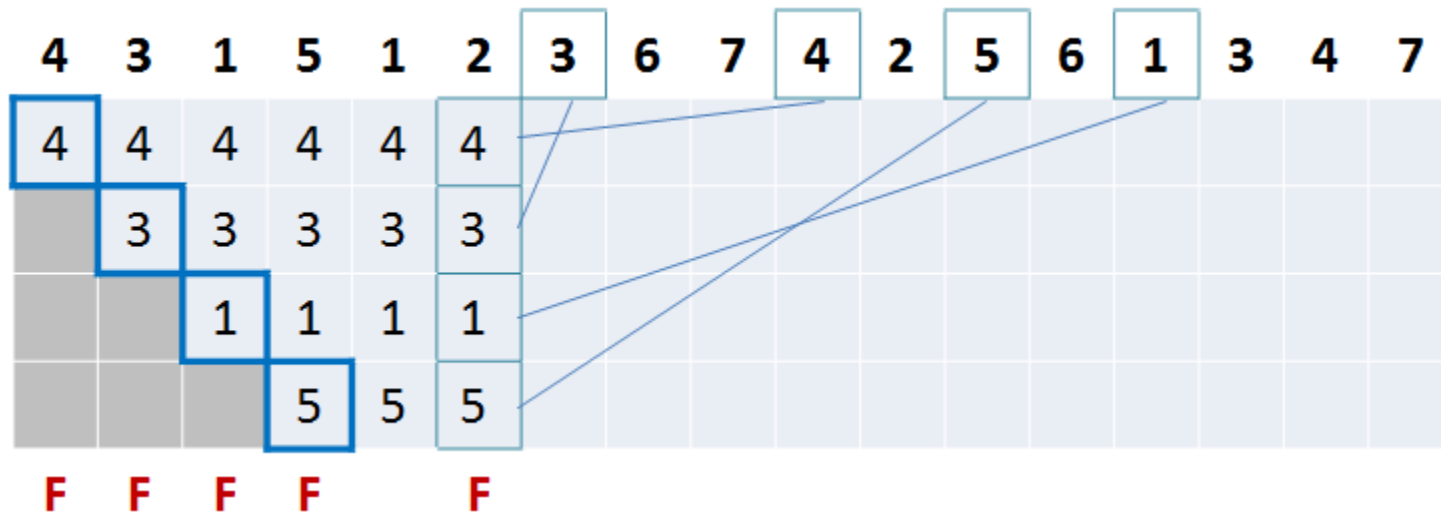
**/2/3/2/1/5/2/4/5/3/2/5/2/**

Τότε θα έχουμε τα παρακάτω σφάλματα σελίδας (σημειώνονται με \*):

**/2\*/3\*/2/1\*/5\*/2/4\*/5/3/2\*/5/2/**



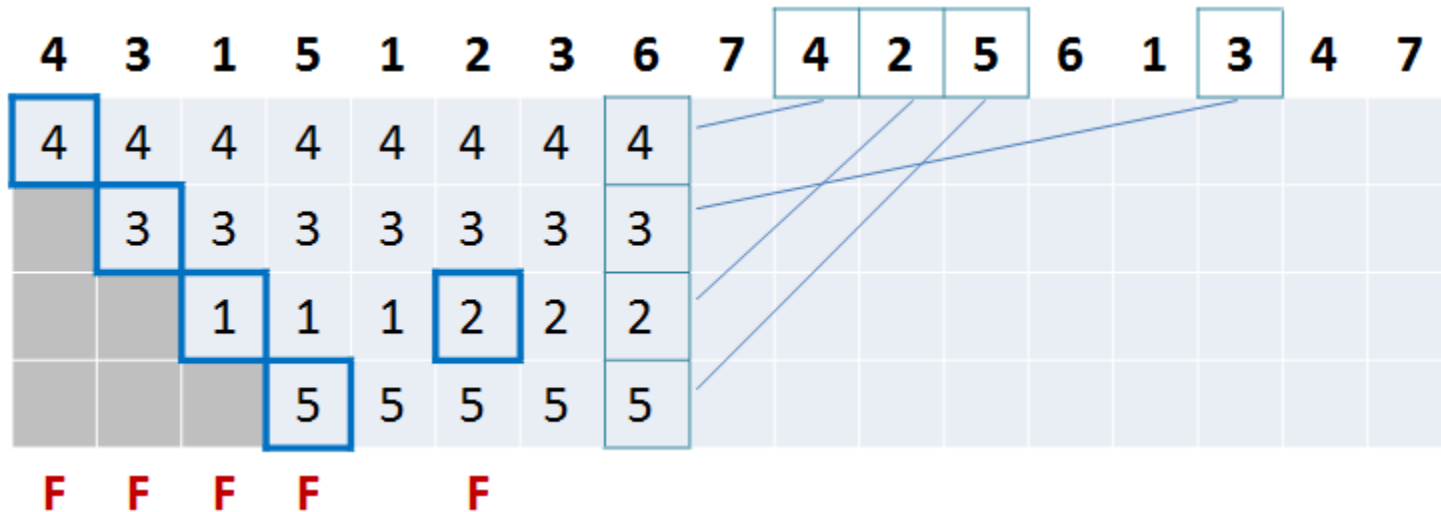
# Παραδείγματα (3/9)



Υποψήφιος σελίδες για να αντικατασταθούν: 4, 3, 1, 5.  
Επιλέχθηκε: 1.



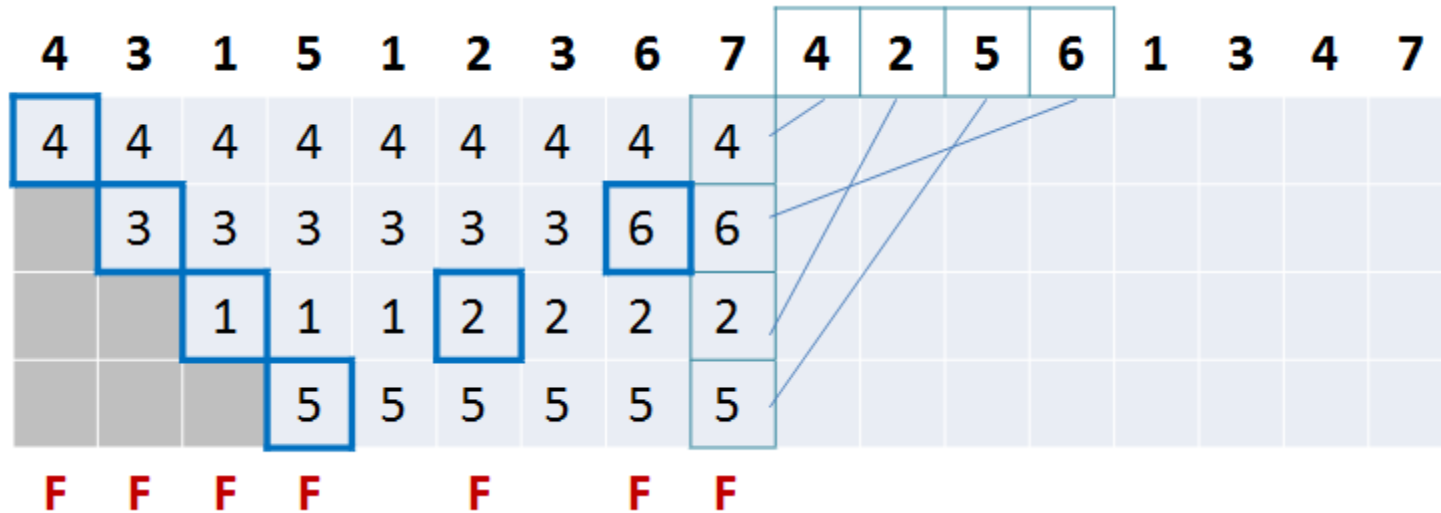
# Παραδείγματα (4/9)



Υποψήφιος σελίδες για να αντικατασταθούν: 4, 3, 2, 5.  
Επιλέχθηκε: 3.



# Παραδείγματα (5/9)



Υποψήφιες σελίδες για να αντικατασταθούν: 4, 6, 2, 5.  
Επιλέχθηκε: 6.



# Παραδείγματα (6/9)

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 3 | 1 | 5 | 1 | 2 | 3 | 6 | 7 | 4 | 2 | 5 | 6 | 1 | 3 | 4 | 7 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
|   | 3 | 3 | 3 | 3 | 3 | 3 | 6 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
|   |   | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
|   |   |   | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| F | F | F | F |   | F |   | F | F |   |   |   |   |   |   |   | F |

**Υποψήφιες σελίδες για να αντικατασταθούν:** 4, 7, 2, 5.

**Επιλέχθηκε:** 2 ή 5 (και οι δύο σελίδες δε θα ξανά χρησιμοποιηθούν).



# Παραδείγματα (7/9)

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 3 | 1 | 5 | 1 | 2 | 3 | 6 | 7 | 4 | 2 | 5 | 6 | 1 | 3 | 4 | 7 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |   |   |
|   | 3 | 3 | 3 | 3 | 3 | 3 | 6 | 7 | 7 | 7 | 7 | 7 | 7 |   |   |   |
|   |   | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 6 | 6 |   |   |   |   |
|   |   |   | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |   |   |   |
| F | F | F | F |   | F |   | F | F |   |   |   | F | F |   |   |   |

**Υποψήφιες σελίδες για να αντικατασταθούν:** 4, 7, 6, 5.

**Επιλέχθηκε:** 5 ή 6 (και οι δύο σελίδες δε θα ξανά χρησιμοποιηθούν).



# Παραδείγματα (8/9)

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 3 | 1 | 5 | 1 | 2 | 3 | 6 | 7 | 4 | 2 | 5 | 6 | 1 | 3 | 4 | 7 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |   |   |
|   | 3 | 3 | 3 | 3 | 3 | 3 | 6 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |   |   |
|   |   | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 6 | 6 | 6 |   |   |   |
|   |   |   | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 1 | 1 |   |   |
| F | F | F | F |   | F |   | F | F |   |   |   | F | F |   |   |   |

**Υποψήφιες σελίδες για να αντικατασταθούν:** 4, 7, 6, 1.

**Επιλέχθηκε:** 1 ή 6 (και οι δύο σελίδες δε θα ξανά χρησιμοποιηθούν).



# Παραδείγματα (9/9)

Έστω ότι έχουμε:

- Επτά σελίδες
- Πέντε πλαίσια

|          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| <b>4</b> | <b>3</b> | <b>1</b> | <b>5</b> | <b>1</b> | <b>2</b> | <b>3</b> | <b>6</b> | <b>7</b> | <b>4</b> | <b>2</b> | <b>5</b> | <b>6</b> | <b>1</b> | <b>3</b> | <b>4</b> | <b>7</b> |
| 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        |
|          | 3        | 3        | 3        | 3        | 3        | 3        | 6        | 7        | 7        | 7        | 7        | 7        | 7        | 7        | 7        | 7        |
|          |          | 1        | 1        | 1        | 2        | 2        | 2        | 2        | 2        | 2        | 2        | 6        | 6        | 3        | 3        | 3        |
|          |          |          | 5        | 5        | 5        | 5        | 5        | 5        | 5        | 5        | 5        | 5        | 1        | 1        | 1        | 1        |
| <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> |          | <b>F</b> |          | <b>F</b> | <b>F</b> |          |          |          | <b>F</b> | <b>F</b> | <b>F</b> |          |          |

**Σφάλματα σελίδας: 10**



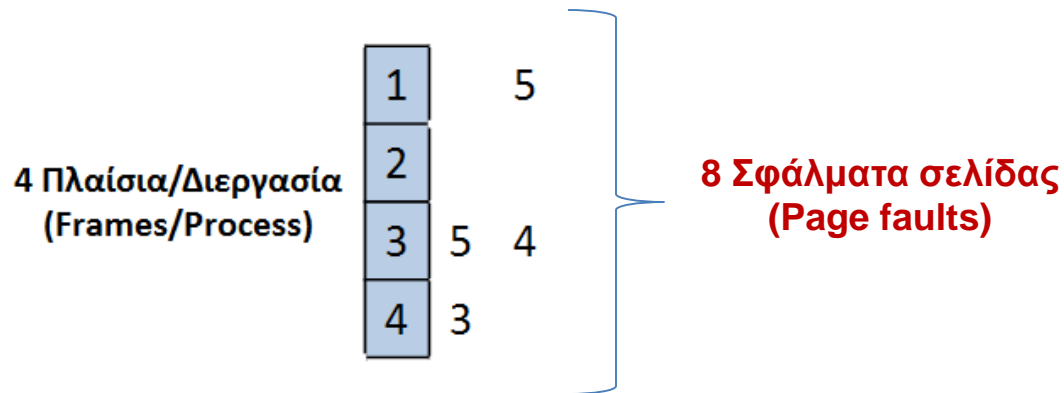
# Λιγότερο πρόσφατα χρησιμοποιούμενη σελίδα (Least Recently Used – LRU)

- Αντικαθιστά εκείνη τη σελίδα στην οποία δεν έγινε αναφορά για το μεγαλύτερο χρονικό διάστημα.
- Σύμφωνα με τον κανόνα της τοπικότητας της αναφοράς αυτή θα είναι η σελίδα με την ελάχιστη πιθανότητα αναφοράς στο εγγύς μέλλον.
- Κάθε σελίδα πρέπει να αποκτά μια ετικέτα με τη χρονική στιγμή κατά την οποία έγινε η τελευταία αναφορά σε αυτήν. Αυτό θα απαιτήσει ένα μεγάλο ποσό επιβάρυνσης.



# Παράδειγμα (1/14)

Αλφαριθμητικό αναφοράς: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5.



# Παραδείγματα (2/14)

Αν η μνήμη αποτελείται από 3 πλαίσια και η διεργασία έχει 5 σελίδες, με την ακόλουθη χρονική σειρά αναφοράς:

**/2/3/2/1/5/2/4/5/3/2/5/2/**

Τότε θα έχουμε τα παρακάτω σφάλματα σελίδας (σημειώνονται με \*):

**/2\*/3\*/2/1\*/5\*/2/4\*/5/3\*/2\*/5/2/**

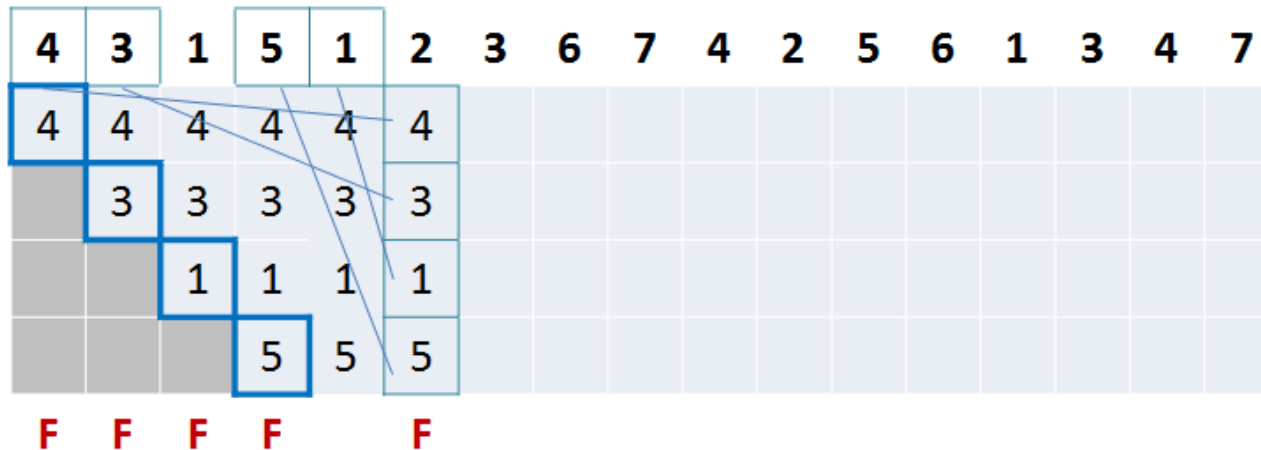


# Παραδείγματα (3/14)

Έστω ότι έχουμε:

- Επτά σελίδες
- Τέσσερα πλαίσια

## Αλφαριθμητικό αναφοράς



Υποψήφιες σελίδες για να αντικατασταθούν: 4, 3, 1, 5.

Επιλέχθηκε: 4





# Παραδείγματα (4/14)

Έστω ότι έχουμε:

- Επτά σελίδες
- Τέσσερα πλαίσια

## Αλφαριθμητικό αναφοράς

|  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | 4 | 3 | 1 | 5 | 1 | 2 | 3 | 6 | 7 | 4 | 2 | 5 | 6 | 1 | 3 | 4 | 7 |
|  | 4 | 4 | 4 | 4 | 4 | 2 | 2 | 2 |   |   |   |   |   |   |   |   |   |
|  |   | 3 | 3 | 3 | 3 | 3 | 3 | 3 |   |   |   |   |   |   |   |   |   |
|  |   |   | 1 | 1 | 1 | 1 | 1 | 1 |   |   |   |   |   |   |   |   |   |
|  |   |   |   | 5 | 5 | 5 | 5 | 5 |   |   |   |   |   |   |   |   |   |
|  | F | F | F | F |   | F |   | F |   |   |   |   |   |   |   |   |   |

Υποψήφιες σελίδες για να αντικατασταθούν: 2, 3, 1, 5.  
Επιλέχθηκε: 5.



# Παραδείγματα (5/14)

Έστω ότι έχουμε:

- Επτά σελίδες
- Τέσσερα πλαίσια

## Αλφαριθμητικό αναφοράς

|  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | 4 | 3 | 1 | 5 | 1 | 2 | 3 | 6 | 7 | 4 | 2 | 5 | 6 | 1 | 3 | 4 | 7 |
|  | 4 | 4 | 4 | 4 | 4 | 2 | 2 | 2 | 2 |   |   |   |   |   |   |   |   |
|  |   | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |   |   |   |   |   |   |   |   |
|  |   |   | 1 | 1 | 1 | 1 | 1 | 1 | 1 |   |   |   |   |   |   |   |   |
|  |   |   |   | 5 | 5 | 5 | 5 | 6 | 6 |   |   |   |   |   |   |   |   |
|  | F | F | F | F |   | F |   | F | F |   |   |   |   |   |   |   |   |

Υποψήφιες σελίδες για να αντικατασταθούν: 2, 3, 1, 6.  
Επιλέχθηκε: 1.



# Παραδείγματα (6/14)

Έστω ότι έχουμε:

- Επτά σελίδες
- Τέσσερα πλαίσια

## Αλφαριθμητικό αναφοράς

|  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | 4 | 3 | 1 | 5 | 1 | 2 | 3 | 6 | 7 | 4 | 2 | 5 | 6 | 1 | 3 | 4 | 7 |
|  | 4 | 4 | 4 | 4 | 4 | 2 | 2 | 2 | 2 | 2 |   |   |   |   |   |   |   |
|  |   | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |   |   |   |   |   |   |   |
|  |   |   | 1 | 1 | 1 | 1 | 1 | 1 | 7 | 7 |   |   |   |   |   |   |   |
|  |   |   |   | 5 | 5 | 5 | 5 | 6 | 6 | 6 |   |   |   |   |   |   |   |
|  | F | F | F | F |   | F |   | F | F | F |   |   |   |   |   |   |   |

Υποψήφιες σελίδες για να αντικατασταθούν: 2, 3, 7, 6.  
Επιλέχθηκε: 2.



# Παραδείγματα (7/14)

Έστω ότι έχουμε:

- Επτά σελίδες
- Τέσσερα πλαίσια

## Αλφαριθμητικό αναφοράς

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 3 | 1 | 5 | 1 | 2 | 3 | 6 | 7 | 4 | 2 | 5 | 6 | 1 | 3 | 4 | 7 |
| 4 | 4 | 4 | 4 | 4 | 2 | 2 | 2 | 2 | 4 | 4 |   |   |   |   |   |   |
|   | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |   |   |   |   |   |   |
|   |   | 1 | 1 | 1 | 1 | 1 | 1 | 7 | 7 | 7 |   |   |   |   |   |   |
|   |   |   | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 6 |   |   |   |   |   |   |
| F | F | F | F |   | F |   | F | F | F | F |   |   |   |   |   |   |

Υποψήφιες σελίδες για να αντικατασταθούν: 4, 3, 7, 6.  
Επιλέχθηκε: 3.



# Παραδείγματα (8/14)

Έστω ότι έχουμε:

- Επτά σελίδες
- Τέσσερα πλαίσια

## Αλφαριθμητικό αναφοράς

|  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | 4 | 3 | 1 | 5 | 1 | 2 | 3 | 6 | 7 | 4 | 2 | 5 | 6 | 1 | 3 | 4 | 7 |
|  | 4 | 4 | 4 | 4 | 4 | 2 | 2 | 2 | 2 | 4 | 4 | 4 |   |   |   |   |   |
|  |   | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 |   |   |   |   |   |
|  |   |   | 1 | 1 | 1 | 1 | 1 | 1 | 7 | 7 | 7 | 7 |   |   |   |   |   |
|  |   |   |   | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 6 | 6 |   |   |   |   |   |
|  | F | F | F | F |   | F |   | F | F | F | F | F |   |   |   |   |   |

Υποψήφιες σελίδες για να αντικατασταθούν: 4, 2, 7, 6.  
Επιλέχθηκε: 6.







# Παραδείγματα (11/14)

Έστω ότι έχουμε:

- Επτά σελίδες
- Τέσσερα πλαίσια

## Αλφαριθμητικό αναφοράς

|  |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
|--|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|  | <b>4</b> | <b>3</b> | <b>1</b> | <b>5</b> | <b>1</b> | <b>2</b> | <b>3</b> | <b>6</b> | <b>7</b> | <b>4</b> | <b>2</b> | <b>5</b> | <b>6</b> | <b>1</b> | <b>3</b> | <b>4</b> | <b>7</b> |
|  | 4        | 4        | 4        | 4        | 4        | 2        | 2        | 2        | 2        | 4        | 4        | 4        | 4        | 1        | 1        |          |          |
|  |          | 3        | 3        | 3        | 3        | 3        | 3        | 3        | 3        | 3        | 2        | 2        | 2        | 2        | 2        |          |          |
|  |          |          | 1        | 1        | 1        | 1        | 1        | 1        | 7        | 7        | 7        | 7        | 6        | 6        | 6        |          |          |
|  |          |          |          | 5        | 5        | 5        | 5        | 6        | 6        | 6        | 6        | 5        | 5        | 5        | 5        |          |          |
|  | <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> |          | <b>F</b> |          | <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> |          |          |

Υποψήφιες σελίδες για να αντικατασταθούν: 1, 2, 6, 5.  
Επιλέχθηκε: 2.





# Παραδείγματα (12/14)

Έστω ότι έχουμε:

- Επτά σελίδες
- Τέσσερα πλαίσια

## Αλφαριθμητικό αναφοράς

|  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | 4 | 3 | 1 | 5 | 1 | 2 | 3 | 6 | 7 | 4 | 2 | 5 | 6 | 1 | 3 | 4 | 7 |
|  | 4 | 4 | 4 | 4 | 4 | 2 | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 1 | 1 | 1 |   |
|  |   | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 3 | 3 |   |
|  |   |   | 1 | 1 | 1 | 1 | 1 | 1 | 7 | 7 | 7 | 7 | 6 | 6 | 6 | 5 |   |
|  |   |   |   | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 6 | 5 | 5 | 5 | 5 | 6 |   |
|  | F | F | F | F |   | F |   | F | F | F | F | F | F | F | F | F |   |

Υποψήφιες σελίδες για να αντικατασταθούν: 1, 3, 6, 5.  
Επιλέχθηκε: 5.



# Παραδείγματα (13/14)

Έστω ότι έχουμε:

- Επτά σελίδες
- Τέσσερα πλαίσια

## Αλφαριθμητικό αναφοράς

|  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | 4 | 3 | 1 | 5 | 1 | 2 | 3 | 6 | 7 | 4 | 2 | 5 | 6 | 1 | 3 | 4 | 7 |
|  | 4 | 4 | 4 | 4 | 4 | 2 | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 1 | 1 | 1 | 1 |
|  |   | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 3 | 3 |   | 3 |
|  |   |   | 1 | 1 | 1 | 1 | 1 | 1 | 7 | 7 | 7 | 7 | 6 | 6 | 6 | 5 | 6 |
|  |   |   |   | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 6 | 5 | 5 | 5 | 5 | 4 | 4 |
|  | F | F | F | F |   | F |   | F | F | F | F | F | F | F | F | F |   |

Υποψήφιες σελίδες για να αντικατασταθούν: 1, 3, 6, 4.  
Επιλέχθηκε: 6.



# Παραδείγματα (14/14)

Έστω ότι έχουμε:

- Επτά σελίδες
- Τέσσερα πλαίσια

## Αλφαριθμητικό αναφοράς

|          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| <b>4</b> | <b>3</b> | <b>1</b> | <b>5</b> | <b>1</b> | <b>2</b> | <b>3</b> | <b>6</b> | <b>7</b> | <b>4</b> | <b>2</b> | <b>5</b> | <b>6</b> | <b>1</b> | <b>3</b> | <b>4</b> | <b>7</b> |
| 4        | 4        | 4        | 4        | 4        | 2        | 2        | 2        | 2        | 4        | 4        | 4        | 4        | 1        | 1        | 1        | 1        |
|          | 3        | 3        | 3        | 3        | 3        | 3        | 3        | 3        | 3        | 2        | 2        | 2        | 2        | 3        | 3        | 3        |
|          |          | 1        | 1        | 1        | 1        | 1        | 1        | 7        | 7        | 7        | 7        | 6        | 6        | 6        | 5        | 7        |
|          |          |          | 5        | 5        | 5        | 5        | 6        | 6        | 6        | 6        | 5        | 5        | 5        | 5        | 4        | 4        |
| <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> |          | <b>F</b> |          | <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> | <b>F</b> |          |

**Σφάλματα σελίδας: 15**



# Άσκηση 1

Αλφαριθμητικό αναφοράς: A, B, C, A, B, D, A, D, B, C, B.

|   | FIFO | OPT | LRU |
|---|------|-----|-----|
| A | A    | A   | A   |
| B | B    | B   | B   |
| C | C    | C   | C   |
| A | A    | A   | A   |
| B | B    | B   | B   |
| D | D    |     |     |
| A | D    |     |     |
| D | D    |     |     |
| B | D    |     |     |
| C | D    |     |     |
| B | C    |     |     |

Σφάλματα:

FIFO: 7

OPT: 5

LRU: 5

Να συμπληρωθούν τα κενά στους πίνακες.



# Άσκηση 2 (1/2)

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων:

**1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.**

Πόσα σφάλματα σελίδας θα γίνουν για τους παρακάτω αλγόριθμους αντικατάστασης, υποθέτοντας **1, 2, 3, 4, 5, 6, 7** πλαίσια; Όλα τα πλαίσια είναι καταρχήν κενά, έτσι οι πρώτες σελίδες θα στοιχίσουν ένα σφάλμα σελίδας.

1. LRU replacement.
2. FIFO replacement.
3. Optimal (βέλτιστη).



# Άσκηση 2 (2/2)

| Πλήθος πλαισίων | LRU | FIFO | Βέλτιστη |
|-----------------|-----|------|----------|
| 1               | 20  | 20   | 20       |
| 2               | 17  | 18   | 15       |
| 3               | 15  | 16   | 11       |
| 4               | 10  | 14   | 8        |
| 5               | 8   | 10   | 7        |
| 6               | 7   | 9    | 7        |
| 7               | 7   | 7    | 7        |



# Άσκηση 3

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων :

**A, B, C, A, D, E, C, B, F, E, A, C**

Θεωρείστε επίσης ότι υπάρχουν 4 πλαίσια, καταρχήν κενά. Πόσα σφάλματα σελίδων θα συμβούν και ποια αν χρησιμοποιηθούν οι παρακάτω αλγόριθμοι αντικατάστασης; Να συμπληρώσετε τους παρακάτω πίνακες:

- Αλγόριθμος LRU (λιγότερο πρόσφατα χρησιμοποιούμενη σελίδα).
- Βέλτιστη πολιτική.



# Άλλες πολιτικές αντικατάστασης σελίδων

## Clock Policy (1/4)

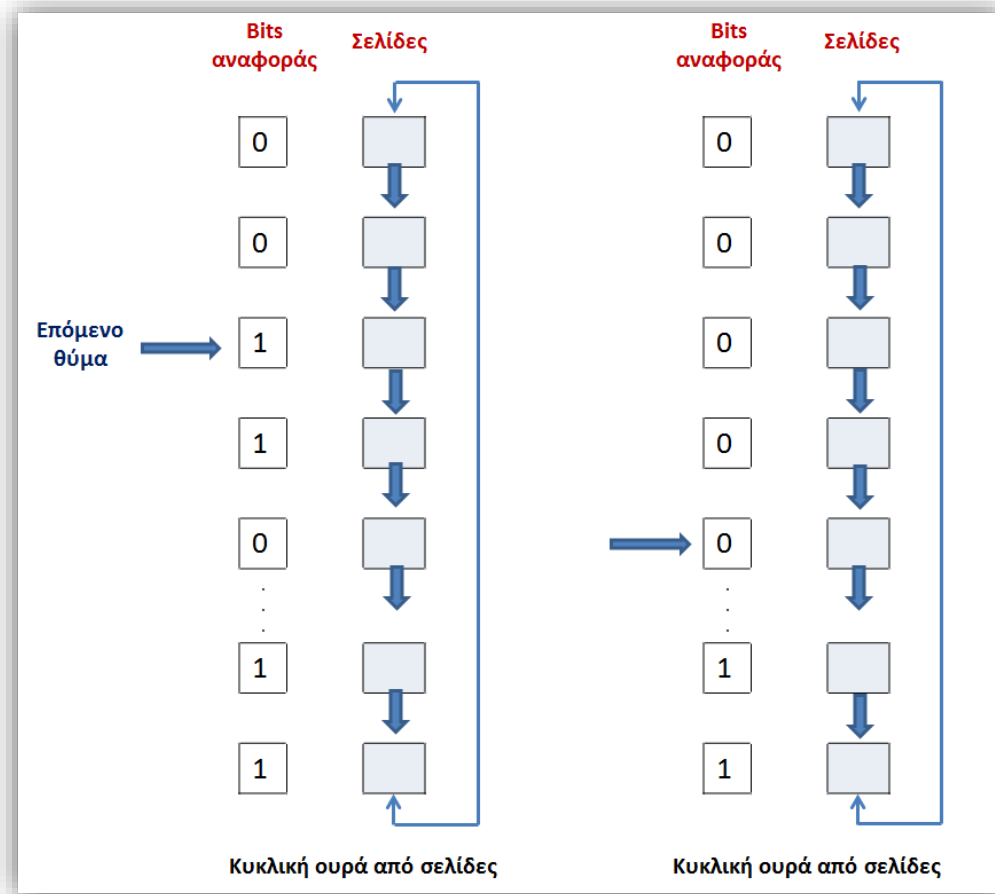
- Χρησιμοποιεί ένα επιπλέον bit, το “use bit”.
- Όταν μια διεργασία φορτώνεται αρχικά στη μνήμη ή αναφέρεται, το use bit έχει την τιμή 1.
- Όταν πρόκειται να αντικατασταθεί μια σελίδα, το Λειτουργικό Σύστημα διατρέχει το σύνολο των frames μετατρέποντας τα 1's σε 0.
- Το πρώτο frame που έχει το use bit ίσο με 0 αντικαθίσταται.





# Άλλες πολιτικές αντικατάστασης σελίδων

## Clock Policy (2/4)



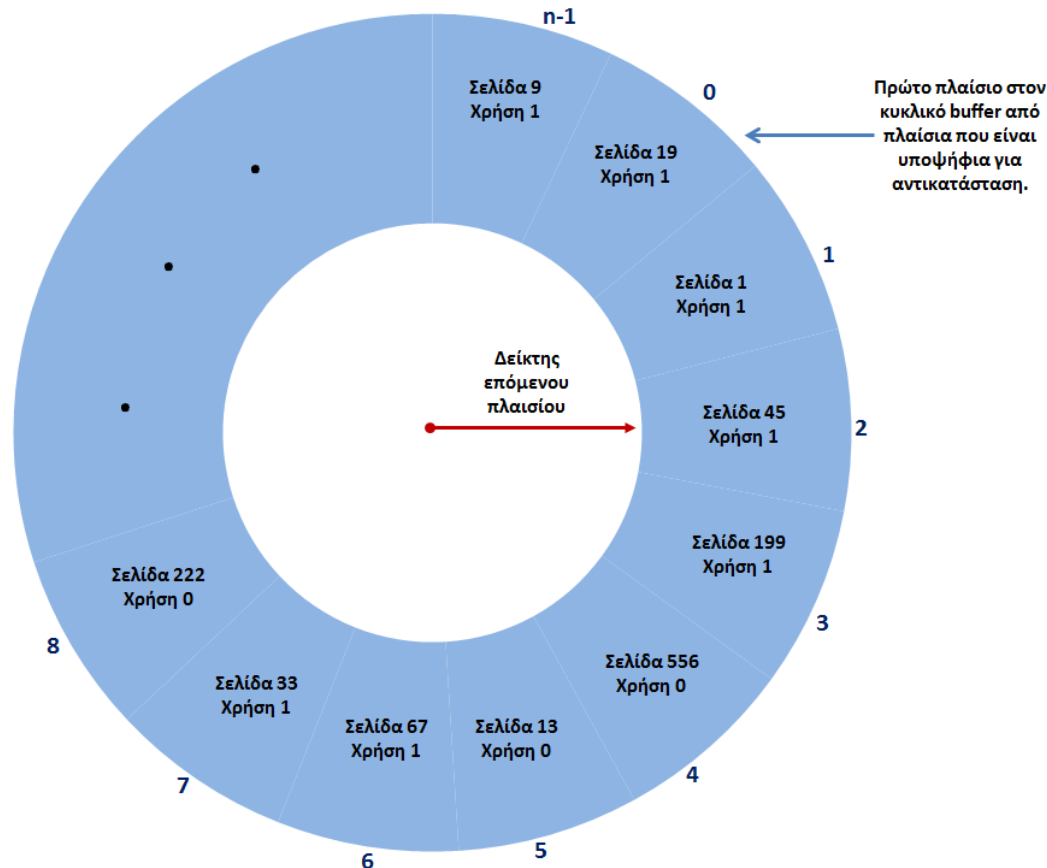
### Πολιτική ρολογιού

Κύρια Μνήμη – Δευτερεύουσα Μνήμη

Τμήμα Πληροφορικής



# Άλλες πολιτικές αντικατάστασης σελίδων Clock Policy (3/4)

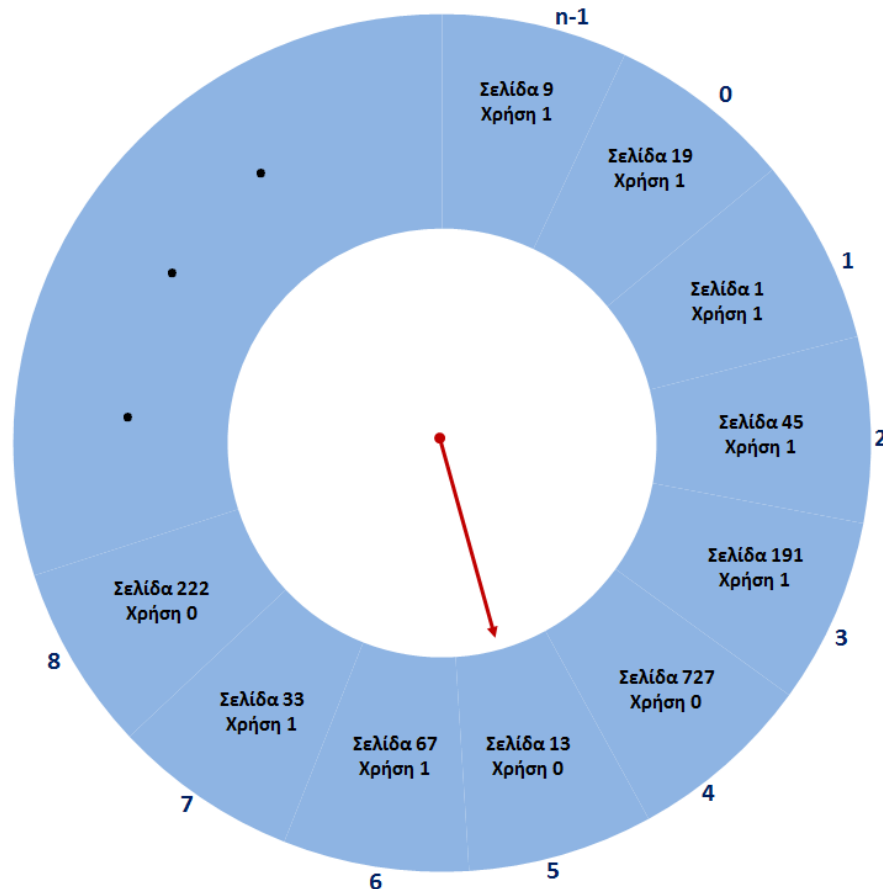


Κατάσταση του buffer ακριβώς πριν την αντικατάσταση σελίδας



# Άλλες πολιτικές αντικατάστασης σελίδων

## Clock Policy (4/4)



Κατάσταση του buffer ακριβώς μετά την επόμενη αντικατάσταση σελίδας



# Συνδυασμένα παραδείγματα

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|
|       |    | 2  | 3  | 2  | 1  | 5  | 2  | 4  | 5  | 3  | 2  | 5  | 2  |
| OPT   | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 4  | 4  | 4  | 2  | 2  | 2  |
|       |    | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  |
|       |    |    |    | 1  | 5  | 5  | 5  | 5  | 5  | 5  | 5  | 5  | 5  |
|       |    |    |    |    |    | F  |    | F  |    |    | F  |    |    |
| LRU   | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 3  | 3  | 3  | 3  |
|       |    | 3  | 3  | 3  | 5  | 5  | 5  | 5  | 5  | 5  | 5  | 5  | 5  |
|       |    |    |    | 1  | 1  | 1  | 4  | 4  | 4  | 2  | 2  | 2  | 2  |
|       |    |    |    |    |    | F  |    | F  |    | F  | F  |    |    |
| FIFO  | 2  | 2  | 2  | 2  | 5  | 5  | 5  | 5  | 3  | 3  | 3  | 3  | 3  |
|       |    | 3  | 3  | 3  | 3  | 2  | 2  | 2  | 2  | 2  | 5  | 5  | 5  |
|       |    |    |    | 1  | 1  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 2  |
|       |    |    |    |    |    | F  | F  | F  |    | F  |    | F  | F  |
| CLOCK | 2* | 2* | 2* | 2* | 5* | 5* | 5* | 5* | 3* | 3* | 3* | 3* | 3* |
|       |    | 3* | 3* | 3* | 3  | 2* | 2* | 2* | 2  | 2* | 2* | 2* | 2* |
|       |    |    |    | 1* | 1  | 1  | 4* | 4* | 4  | 4  | 5* | 5* | 5* |
|       |    |    |    |    |    | F  | F  | F  |    | F  |    | F  |    |

Συμπεριφορά τεσσάρων αλγορίθμων αντικατάστασης σελίδας

# Το παράδοξο του Belady

- Συμβολοσειρά αναφοράς
  - 1,2,3,4,1,2,5,1,2,3,4,5
- Με 3 πλαίσια έχουμε 9 σφάλματα σελίδας.
- Με 4 πλαίσια έχουμε 10 σφάλματα σελίδας.

Υπάρχει περίπτωση η αύξηση του αριθμού των πλαισίων να αυξήσει και τον αριθμό σφαλμάτων (πολύ σπάνια).



# Σφάλμα σελίδας και μέγεθος σελίδας (1/3)

- Μικρότερο μέγεθος σελίδας: λιγότερο ποσό εσωτερικού κατακερματισμού.
- Μικρότερο μέγεθος σελίδας: απαιτούνται περισσότερες σελίδες ανά διεργασία.
- Περισσότερες σελίδες ανά διεργασία σημαίνει μεγαλύτερους πίνακες σελίδων.
- Μεγαλύτεροι πίνακες σελίδων σημαίνει μεγαλύτερο μέρος των πινάκων σελίδων στην εικονική μνήμη.
- Η δευτερεύουσα μνήμη σχεδιάζεται έτσι ώστε να μεταφέρει αποτελεσματικά μεγάλα blocks δεδομένων, έτσι ένα μεγάλο μέγεθος σελίδας είναι καλύτερο.



# Σφάλμα σελίδας και μέγεθος σελίδας (2/3)

- **Μικρό μέγεθος σελίδας:** μεγάλος αριθμός σελίδων θα βρίσκονται στην κύρια μνήμη.
- Με την πρόοδο του χρόνου κατά τη διάρκεια της εκτέλεσης, οι σελίδες στη μνήμη θα περιέχουν τμήματα της διεργασίας που είναι κοντά στις πρόσφατες αναφορές. Τα σφάλματα σελίδας είναι λίγα.
- **Αυξανόμενο μέγεθος σελίδας** οδηγεί σε σελίδες που περιέχουν θέσεις που βρίσκονται πιο μακριά από τις πρόσφατες αναφορές. Τα σφάλματα σελίδας αυξάνονται.



# Σφάλμα σελίδας και μέγεθος σελίδας (3/3)

- Τα πολλαπλά μεγέθη σελίδων παρέχουν την ευελιξία που απαιτείται για την αποτελεσματική χρήση του TLB.
- Μεγάλες σελίδες μπορούν να χρησιμοποιηθούν για εντολές προγράμματος και μικρές για νήματα (threads).
- Τα περισσότερα Λειτουργικά Συστήματα υποστηρίζουν μόνο ένα μέγεθος σελίδας.





# Στρατηγικές αντικατάστασης σελίδων και ανάθεσης πλαισίων

Η αξιολόγηση των αλγορίθμων αντικατάστασης σελίδων βασίζεται :

- Στη σύγκριση των σφαλμάτων που προκύπτουν από την εφαρμογή κάθε αλγορίθμου σε ένα αλφαριθμητικό αναφοράς (**reference string**), που είναι μια ακολουθία αναφοράς σελίδων, για μια τυχαία και αυθαίρετα επιλεγμένη διεργασία.
- Στο πλήθος των πλαισίων που ανατίθενται στη διεργασία από το Λειτουργικό Σύστημα (το πλήθος των σφαλμάτων συνδέεται άμεσα με το πλήθος των παραχωρούμενων πλαισίων).

Στα περισσότερα πραγματικά συστήματα, το Λειτουργικό Σύστημα πρέπει να εφαρμόσει μια στρατηγική που θα πρέπει να αποδίδει καλά σε ένα πλήθος διαφορετικών διεργασιών και με ένα περιορισμένο (λόγω του μεγέθους της κεντρικής μνήμης) πλήθος πλαισίων.



# Αντικατάσταση σελίδων

- Κατά τη σχεδίαση ενός συστήματος σελίδων ιδεατής μνήμης, ο σχεδιαστής του Λειτουργικού Συστήματος πρέπει να αποφασίσει αν οι στρατηγικές αντικατάστασης σελίδων θα εφαρμόζονται σε όλες τις διεργασίες (ολικές στρατηγικές) ή αν κάθε διεργασία θα θεωρείται ξεχωριστή (τοπικές στρατηγικές).
- Οι ολικές στρατηγικές αντικατάστασης τείνουν να αγνοούν τα χαρακτηριστικά της συμπεριφοράς κάθε μιας διεργασίας.
- Οι τοπικές στρατηγικές αντικατάστασης είναι σχεδιασμένες ώστε να παραχωρούν σε κάθε διεργασία μνήμη ανάλογη με τη σπουδαιότητά της και με τρόπο που θα βελτιώνεται η συνολική απόδοση του συστήματος.



# Πολιτική εκκαθάρισης (Cleaning Policy) (1/2)

- Μια πολιτική εκκαθάρισης ορίζει το πότε μια σελίδα που έχει μεταβληθεί θα “γραφεί” στη δευτερεύουσα μνήμη.
- Εκκαθάριση κατά απαίτηση (**Demand cleaning**).
  - Μια σελίδα γράφεται μόνο όταν επιλεγεί για αντικατάσταση.
- Προ-εκκαθάριση.
  - Οι σελίδες γράφονται κατά δέσμες.



# Πολιτική εκκαθάρισης (Cleaning Policy) (2/2)

- Η πιο αποτελεσματική μέθοδος βασίζεται σε [page buffering](#).
- Οι σελίδες που αντικαθίστανται τοποθετούνται σε δυο λίστες.
  - Μεταβαλλόμενες και μη μεταβαλλόμενες.
- Οι σελίδες στη λίστα των μεταβαλλόμενων γράφονται περιοδικά και κατά δεσμίδες.
- Οι σελίδες στη λίστα των μη μεταβαλλόμενων, είτε επανακτούνται όταν ζητηθούν, είτε χάνονται όταν το frame έχει ανατεθεί σε άλλη σελίδα.



# Έλεγχος φόρτωσης (Load control)

- Ορίζει τον αριθμό των διεργασιών που θα βρίσκονται στην κύρια μνήμη.
  - Επίπεδο πολύ-προγραμματισμού.
- Λίγες διεργασίες: πολύς χρόνος θα σπαταλάται λόγω swapping.
- Πολλές διεργασίες: thrashing.

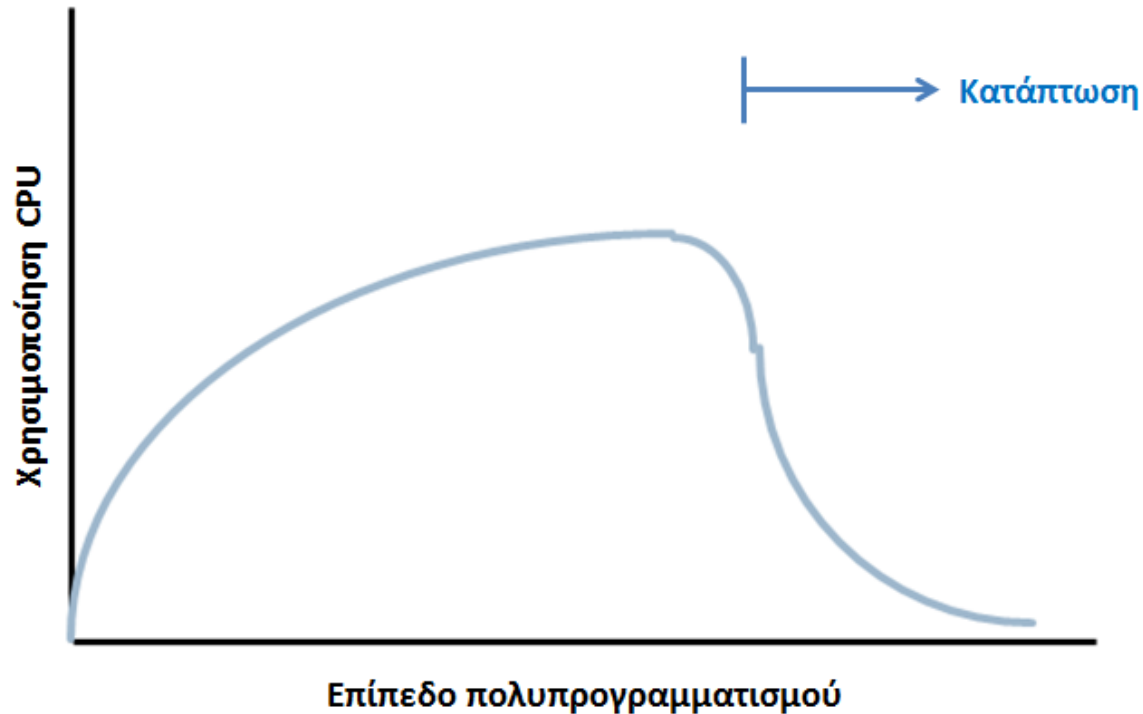


# Το φαινόμενο της κατάπτωσης της ιδεατής μνήμης (Thrashing)

- Αν μια διεργασία δε διαθέτει αρκετές σελίδες θα παράγει πληθώρα σφαλμάτων σελίδων.
  - Η σελίδα που αντικαθίσταται ίσως χρειαστεί και πάλι πολύ σύντομα.
- **Συνέπεια:** ο βαθμός χρήσης της CPU μπορεί να μειωθεί δραματικά.
- Ορισμένα Λειτουργικά Συστήματα εξισορροπούσαν αυτή τη μείωση του βαθμού χρήσης της CPU με την είσοδο περισσότερων διεργασιών.
- Αυτό όμως είχε ως αποτέλεσμα κάθε διεργασία να διαθέτει ακόμη λιγότερες σελίδες και να παράγει περισσότερα σφάλματα και τελικά να προκύπτουν σοβαρότατα προβλήματα απόδοσης.
- **Ο επεξεργαστής σπαταλά το μεγαλύτερο μέρος του χρόνου για την εναλλαγή τμημάτων παρά για την εκτέλεση των εντολών του χρήστη (φαινόμενο κατάπτωσης της ιδεατής μνήμης).**



# Διάγραμμα κατάπτωσης



Διάγραμμα κατάπτωσης



# Αναφορές

- [1]. Stallings William, “Operating systems: Internal and Design Principles”, Fourth edition, Publishing as Prentice Hall, 2000.
- [2]. A.S.Tanenbaum, "Σύγχρονα Λειτουργικά Συστήματα" Τόμος Α', Εκδόσεις, Παπασωτηρίου.
- [3]. H.M. Deitel, "Operating Systems", 2nd edition, Addison-Wesley Publishing Company.
- [3]. W. Stallings, “Λειτουργικά Συστήματα Αρχές Σχεδίασης”, 6η έκδοση, ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ, 2009, Θεσσαλονίκη.







# Τέλος Ενότητας



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

