

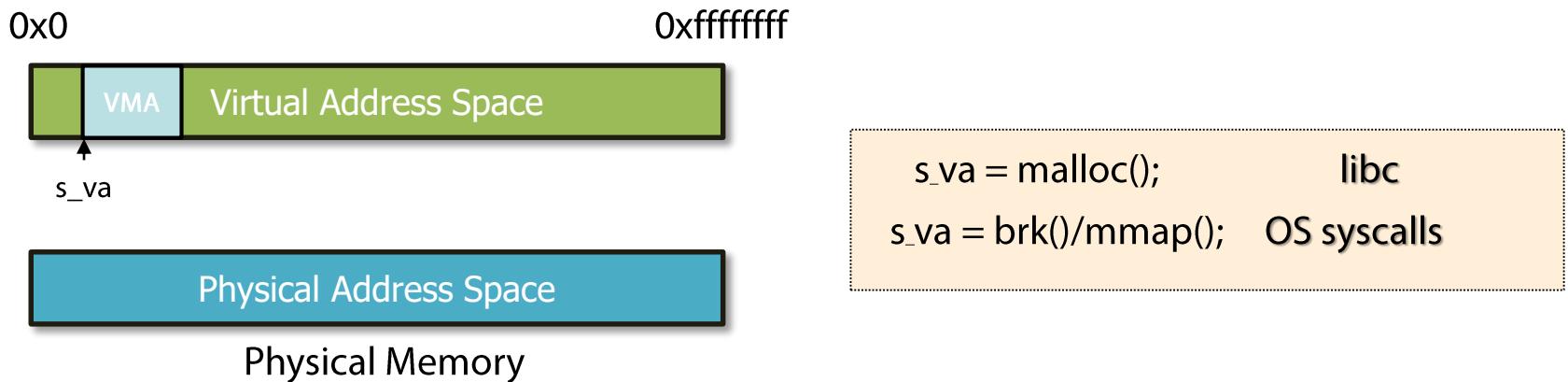


Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Ηλεκτρολόγων Μηχ. και Μηχανικών Υπολογιστών  
Εργαστήριο Υπολογιστικών Συστημάτων

# Εικονική Μνήμη (1/2)

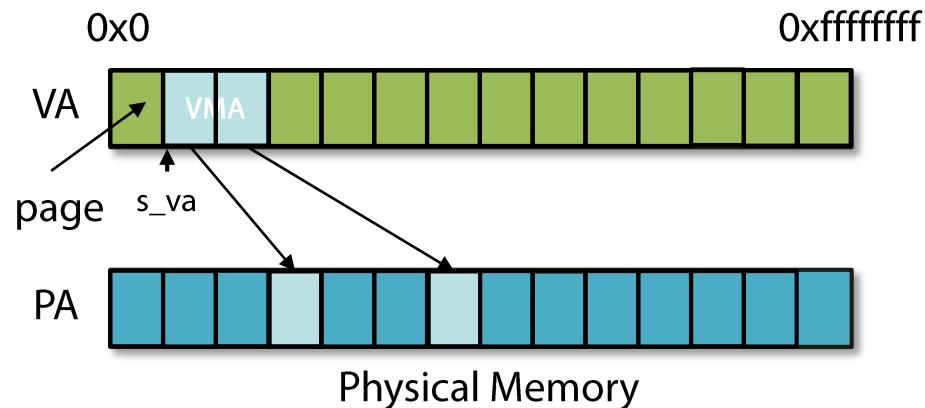
Λειτουργικά Συστήματα Υπολογιστών  
6ο Εξάμηνο, 2021-2022

# Εικονική Μνήμη Επισκόπηση



- ◆ Εικονική μνήμη: γραμμικός συνεχής χώρος διευθύνσεων
- ◆ Κλήσεις συστήματος → το ΛΣ δεσμεύει μια περιοχή εικονικών διευθύνσεων (virtual memory area – VMA)

# Εικονική Μνήμη Επισκόπηση



s\_va = malloc();                            libc  
s\_va = brk()/mmap();    OS syscalls

- ◆ Το ΛΣ διαχειρίζεται τη φυσική μνήμη και τις απεικονίσεις (VA-to-PA mappings)
  - ➡ σε επίπεδο σελίδων (σελιδοποίηση – paging)
  - ➡ σε επίπεδο τμημάτων (κατατμηση – segmentation)

# Εικονική Μνήμη - Σύνοψη

- ◆ Σελιδοποίηση
- ◆ Εικονική Μνήμη και memory overcommit
- ◆ Δυναμική σύνδεση με μοιραζόμενες βιβλιοθήκες
- ◆ Σελιδοποίηση κατ' απαίτηση
- ◆ Σφάλμα σελίδας
  - Χειρισμός από το ΛΣ
- ◆ Δημιουργία διεργασιών σε συστήματα Εικονικής Μνήμης
  - Αντιγραφή κατά την Εγγραφή (Copy-on-Write)
- ◆ Απεικόνιση Αρχείων στην Εικονική Μνήμη

# Εικονική Μνήμη - Σύνοψη

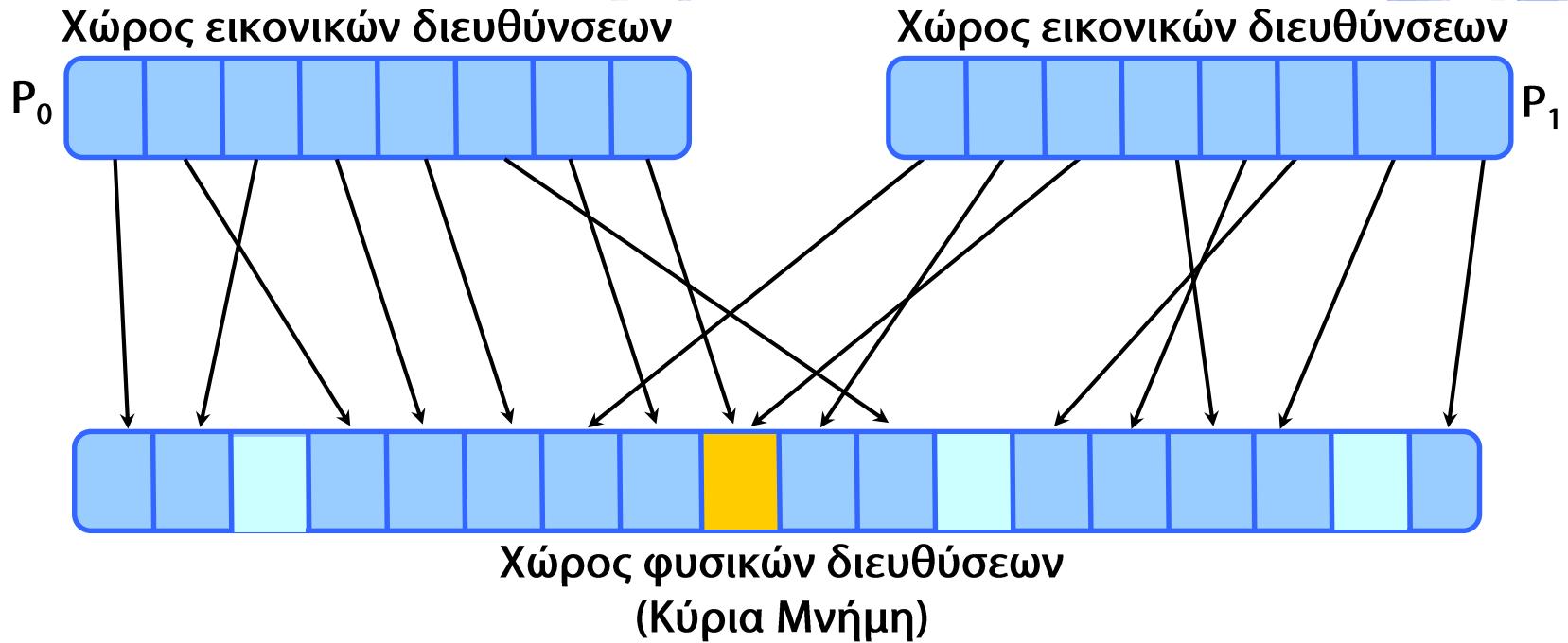
## ◆ Σελιδοποίηση

- ◆ Εικονική Μνήμη και memory overcommit
- ◆ Δυναμική σύνδεση με μοιραζόμενες βιβλιοθήκες
- ◆ Σελιδοποίηση κατ' απαίτηση
- ◆ Σφάλμα σελίδας
  - Χειρισμός από το ΛΣ
- ◆ Δημιουργία διεργασιών σε συστήματα Εικονικής Μνήμης
  - Αντιγραφή κατά την Εγγραφή (Copy-on-Write)
- ◆ Απεικόνιση Αρχείων στην Εικονική Μνήμη

# Σελιδοποίηση (1)

- ◆ Ο χώρος φυσικών διευθύνσεων χωρίζεται σε **πλαίσια** σταθερού μεγέθους (π.χ., 4096 bytes)
- ◆ Ο χώρος λογικών / εικονικών διευθύνσεων χωρίζεται σε **σελίδες**, ίδιου μεγέθους με τα πλαίσια
- ◆ Κάθε σελίδα αντιστοιχίζεται σε οποιοδήποτε πλαίσιο
  - χωρίς περιορισμό συνεχόμενης αποθήκευσης
  - η διεργασία ζει σε διάσπαρτα φυσικά τμήματα
  - αλλά σε γραμμικό χώρο εικονικών διευθύνσεων
- ◆ Το **Υλικό (MMU)** αναλαμβάνει τη μετάφραση
- ◆ Ο πιο συχνός τρόπος μετάφρασης διευθύνσεων

# Σελιδοποίηση (2)



- ◆ Διακριτοί, γραμμικοί χώροι εικονικών διευθύνσεων
- ◆ Προστασία μνήμης
  - Μια διεργασία δεν μπορεί καν να αναφερθεί σε ξένες διευθύνσεις
- ◆ Μοιραζόμενη μνήμη, με αντιστοίχιση στο ίδιο πλαίσιο
- ◆ Δεν έχει εξωτερικό κατακερματισμό

# Σελιδοποίηση (3)

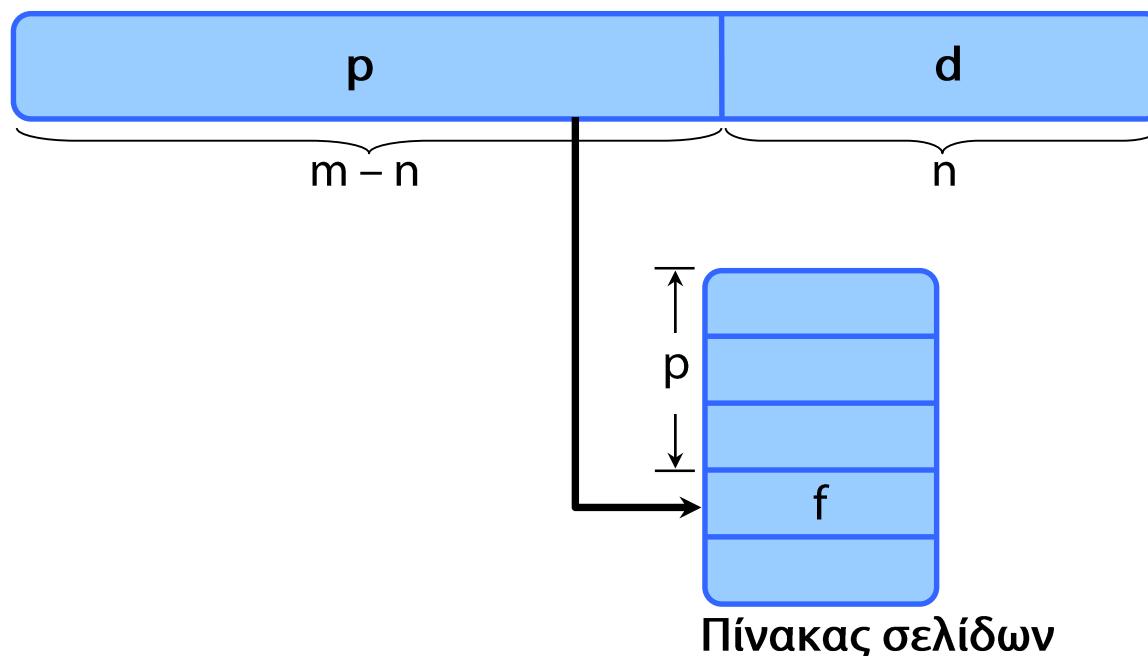
- ◆ Μετάφραση διευθύνσεων με πίνακες σελίδων
  - ➡ Τηρούνται από το ΛΣ, τους συμβουλεύεται το υλικό
- ◆ Μέγεθος σελίδας; 4 KB – 16MB
  - ➡ Μεγάλο μέγεθος → μικρό κόστος διαχείρισης
    - μικρότεροι πίνακες σελίδων, TLB performance
    - εντονότερος εσωτερικός κατακερματισμός
  - ➡ Μικρό μέγεθος
    - αποφεύγεται ο εσωτερικός κατακερματισμός
    - μεγαλύτερο κόστος διαχείρισης

# Σελιδοποίηση - Μετάφραση διευθύνσεων

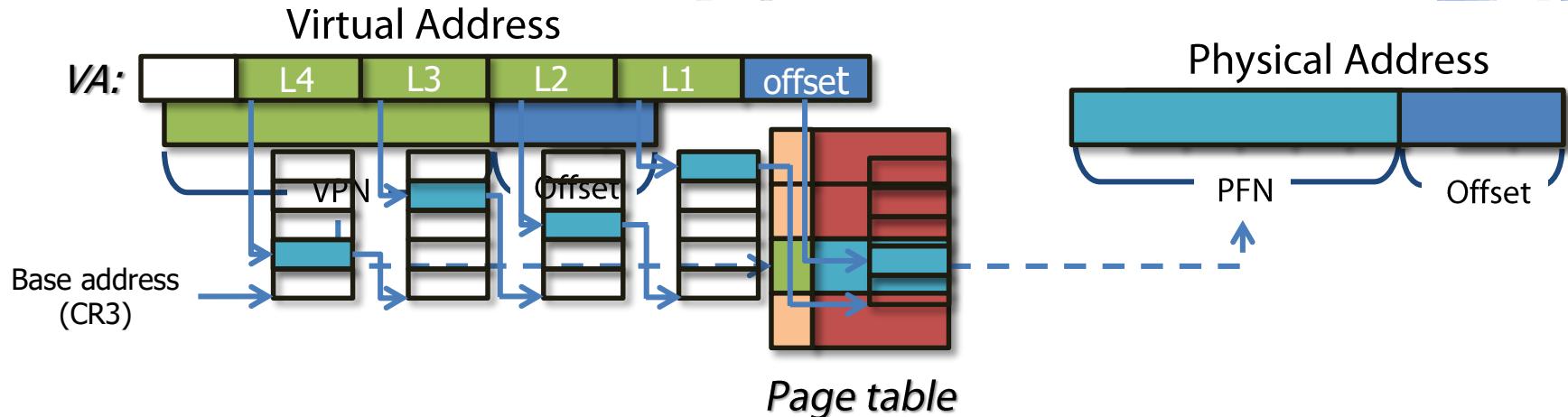
- ◆ Μέγεθος σελίδας  $2^n$
- ◆ Διεύθυνση των  $m$  bits, χώρος εικονικών διεύθυνσεων  $2^m$
- ◆ *Εικονική διεύθυνση*: αριθμός σελίδας + μετατόπιση

Αριθμός εικονικής σελίδας  
(virtual page number)

μετατόπιση (offset)



# Σελιδοποίηση - Μετάφραση διευθύνσεων



- ◆ Μετάφρασεις εικονικών διευθύνσεων σε φυσικές
  - ➡ Πίνακες σελίδων
    - x86: πολυεπίπεδοι (δενδρική δομή radix tree)
  - ➡ Τους θέτει/διατηρεί το ΛΣ, τους διαβάζει/περπατάει το υλικό για να κάνει μεταφράσεις (MMU – page walk)

# Σελιδοποίηση – Προστασία (1)

- ◆ Επιβολή δικαιωμάτων πρόσβασης
  - ➡ με bits προστασίας ανά σελίδα εικονικής μνήμης
- ◆ Π.χ., για μοιραζόμενη μνήμη
  - ➡ Μία διεργασία γράφει, οι άλλες μόνο διαβάζουν
- ◆ Bits πρόσβασης
  - ➡ Read, Write, eXecute
- ◆ Bit εγκυρότητας
  - ➡ Valid, Invalid
- ◆ Σε περίπτωση μη επιτρεπόμενης πρόσβασης;
  - ➡ Trap! Εξαίρεση σελίδας – Page fault

# Σελιδοποίηση – Προστασία (2)

σελίδα 0  
σελίδα 1  
σελίδα 2  
σελίδα 3  
σελίδα 4

Εικονική  
Μνήμη

|   |      |   |
|---|------|---|
| 2 | r, x | v |
| 5 | r, x | v |
| 1 | r    | v |
| 7 | r, w | v |
| 0 |      | i |

Πίνακας  
σελίδων

0  
1 σελίδα 2  
2 σελίδα 0  
3  
4  
5 σελίδα 1  
6  
7 σελίδα 3  
8

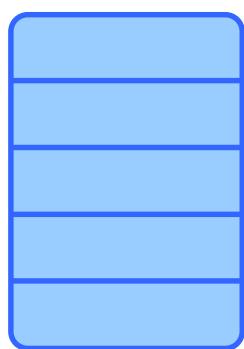
Φυσική  
Μνήμη

- ◆ Ποιες σελίδες είναι κειμένου-κώδικα, ποιες δεδομένων;

# Εικονική Μνήμη - Σύνοψη

- ◆ Σελιδοποίηση
- ◆ Εικονική Μνήμη και *memory overcommit*
- ◆ Δυναμική σύνδεση με μοιραζόμενες βιβλιοθήκες
- ◆ Σελιδοποίηση κατ' απαίτηση
- ◆ Σφάλμα σελίδας
  - Χειρισμός από το ΛΣ
- ◆ Δημιουργία διεργασιών σε συστήματα Εικονικής Μνήμης
  - Αντιγραφή κατά την Εγγραφή (Copy-on-Write)
- ◆ Απεικόνιση Αρχείων στην Εικονική Μνήμη

# Virtual Memory – Εικονική Μνήμη



Εικονική  
Μνήμη



Φυσική  
Μνήμη

# Virtual Memory – Εικονική Μνήμη



Μέγιστη  
Εικονική  
Μνήμη

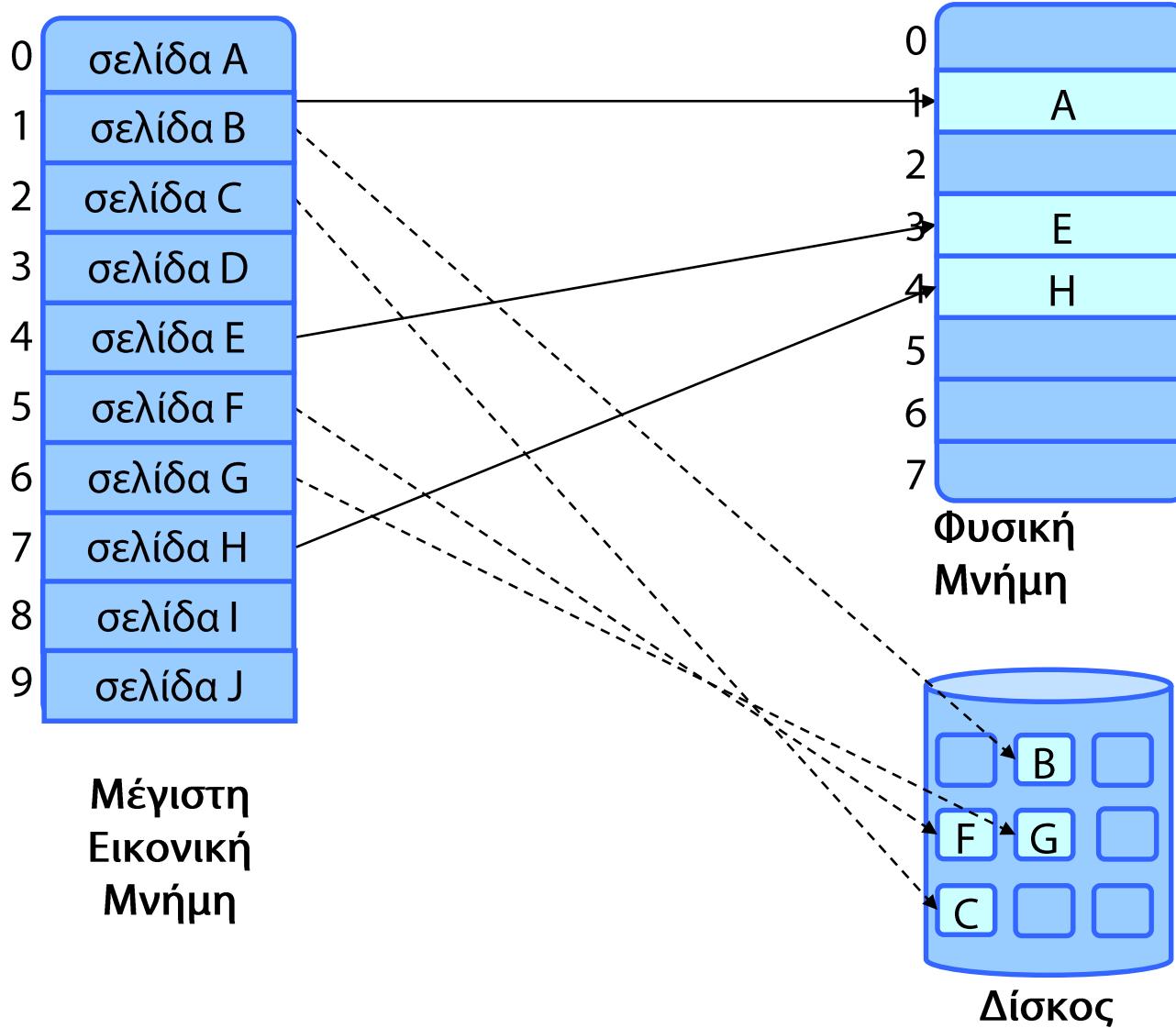


Φυσική  
Μνήμη

# Virtual Memory – Εικονική Μνήμη

- ◆ Λογική μνήμη διαχωρισμένη από τη φυσική μνήμη
  - ➡ ανεξαρτησία από το μέγεθος της φυσικής μνήμης
  - ➡ χώρος λογικών διευθύνσεων πολύ μεγαλύτερος από χώρο φυσικών διευθύνσεων
  - ➡ Τα υπό εκτέλεση προγράμματα βρίσκονται μερικώς στην κύρια μνήμη
    - τα υπόλοιπα δεδομένα που βρίσκονται;

# Memory Overcommitment!



# Virtual Memory – Εικονική Μνήμη

- ◆ Λογική μνήμη διαχωρισμένη από τη φυσική μνήμη
  - ➡ ανεξαρτησία από το μέγεθος της φυσικής μνήμης
  - ➡ χώρος λογικών διευθύνσεων πολύ μεγαλύτερος από χώρο φυσικών διευθύνσεων
  - ➡ Τα υπό εκτέλεση προγράμματα βρίσκονται μερικώς στην κύρια μνήμη
    - τα υπόλοιπα δεδομένα που βρίσκονται;
- ◆ Γλοποίηση:
  - ➡ Κατ' απαίτηση σελιδοποίηση (demand paging)
  - ➡ Κατ' απαίτηση κατάτμηση (demand segmentation)

# Βασικές δομές

- ◆ **Πίνακας σελίδων:**
  - ➔ Περιλαμβάνει τις μεταφράσεις των διευθύνσεων για τις λογικές διευθύνσεις σελίδων που βρίσκονται στη μνήμη.
  - ➔ Τον συμβουλεύεται ο επεξεργαστής (MMU)
  - ➔ Περιλαμβάνει πληροφορία για δικαιώματα πρόσβασης (bit εγκυρότητας, bit πρόσβασης).
  - ➔ Το ΛΣ συντηρεί τον πίνακα σελίδων κάθε διεργασίας.
- ◆ **Χάρτης μνήμης:**
  - ➔ Περιλαμβάνει όλη την πληροφορία για τη μνήμη κάθε διεργασίας
  - ➔ Πληροφορίες και για τις λογικές σελίδες που τα δεδομένα που απεικονίζουν βρίσκονται στο δίσκο

# Πίνακας σελίδων!

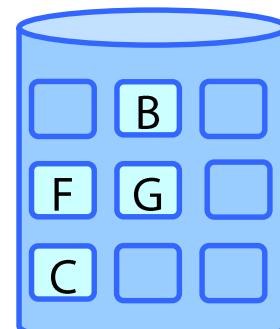
|   |          |
|---|----------|
| 0 | σελίδα Α |
| 1 | σελίδα Β |
| 2 | σελίδα Κ |
| 3 | σελίδα Δ |
| 4 | σελίδα Ε |
| 5 | σελίδα Φ |
| 6 | σελίδα Γ |
| 7 | σελίδα Η |
| 8 | σελίδα Ι |
| 9 | σελίδα Ι |



Πίνακας Σελίδων

|   |   |
|---|---|
| 0 |   |
| 1 | A |
| 2 |   |
| 3 | E |
| 4 | H |
| 5 |   |
| 6 |   |
| 7 |   |

Φυσική  
Μνήμη



Δίσκος

Μέγιστη  
Εικονική  
Μνήμη

# Χάρτης μνήμης!

|   |          |
|---|----------|
| 0 | σελίδα Α |
| 1 | σελίδα Β |
| 2 | σελίδα Κ |
| 3 | σελίδα Δ |
| 4 | σελίδα Ε |
| 5 | σελίδα Φ |
| 6 | σελίδα Γ |
| 7 | σελίδα Η |
| 8 | σελίδα Ι |
| 9 | σελίδα Ι |

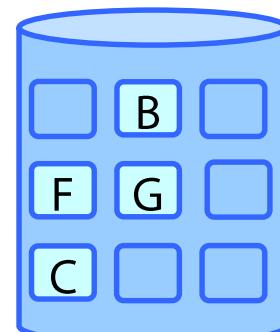
Μέγιστη  
Εικονική  
Μνήμη

|   |   |   |
|---|---|---|
| m | 1 |   |
| d |   | 1 |
| d |   | 6 |
| - | - | - |
| m | 3 |   |
| d |   | 3 |
| d |   | 4 |
| - | - | - |
| m | 4 |   |
| n | - | - |
| - | - | - |

Χάρτης  
μνήμης  
(δομή ΛΣ)

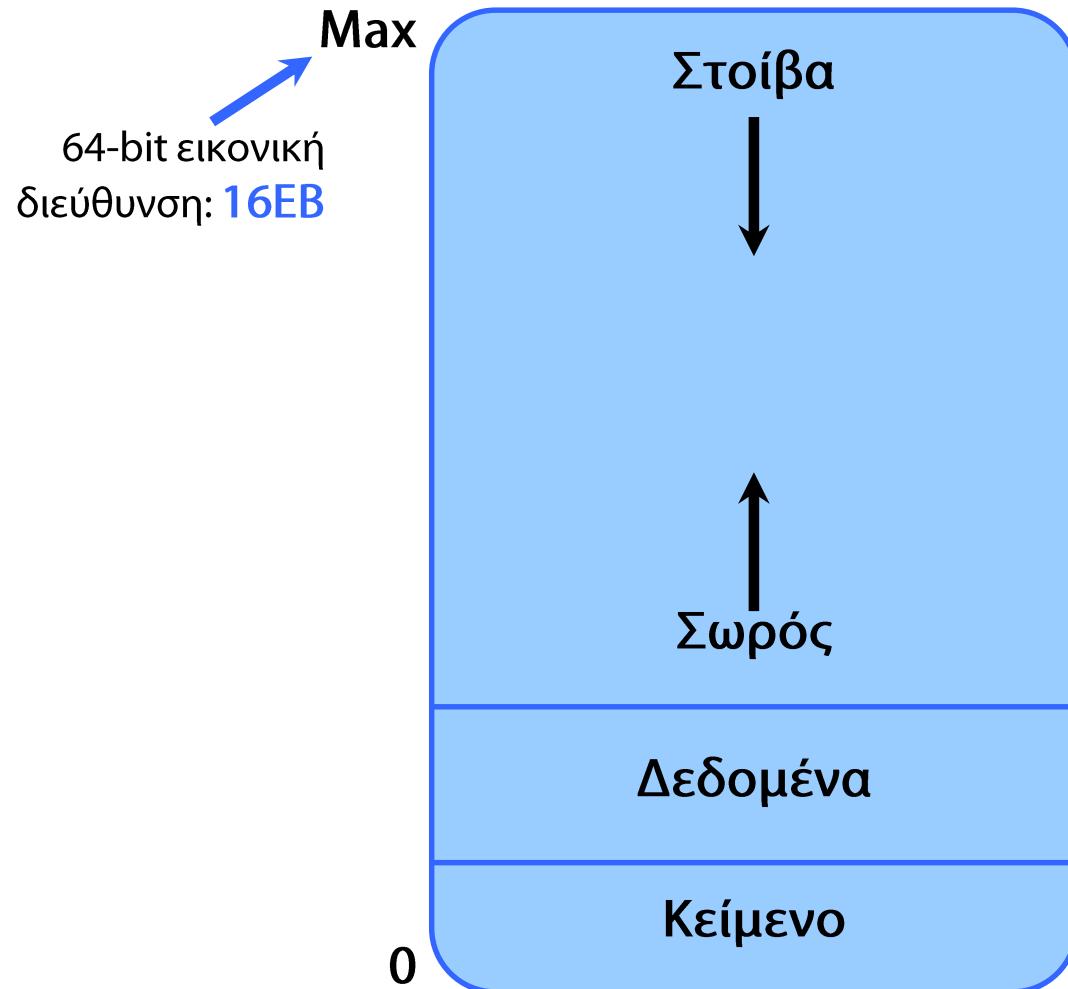
|   |   |
|---|---|
| 0 |   |
| 1 | A |
| 2 |   |
| 3 | E |
| 4 | H |
| 5 |   |
| 6 |   |
| 7 |   |

Φυσική  
Μνήμη



Δίσκος

# Χώρος Εικονικών Διευθύνσεων



# Χάρτης μνήμης (again)

- ◆ Ο χάρτης μνήμης δημιουργείται με τη γέννηση της διεργασίας
- ◆ Το ΛΣ διαβάζει πληροφορίες από το εκτελέσιμο αρχείο και παραχωρεί κατάλληλο χώρο για τον κώδικα, τα δεδομένα και έναν αρχικό χώρο για τη στοίβα και το σωρό
- ◆ Μία σελίδα στο χάρτη μνήμης μπορεί να βρίσκεται στη μνήμη, το δίσκο ή πουθενά (βλ. malloc)
- ◆ Ο χάρτης μνήμης ανανεώνεται αν:
  - ➡ Το πρόγραμμα ζητήσει να μεγαλώσει τη στοίβα του (είτε ρητά με system call είτε μέσω function calls, π.χ. με αναδρομικές κλήσεις)
  - ➡ Το πρόγραμμα ζητήσει/ελευθερώσει δυναμικά μνήμη στο σωρό (malloc/free)
  - ➡ Το πρόγραμμα απεικονίσει ένα αρχείο στη μνήμη (βλ. σύνέχεια)

# Εικονική Μνήμη - Σύνοψη

- ◆ Σελιδοποίηση
- ◆ Εικονική Μνήμη με σελιδοποίηση
- ◆ Δυναμική σύνδεση με μοιραζόμενες βιβλιοθήκες
- ◆ Σελιδοποίηση κατ' απαίτηση
- ◆ Σφάλμα σελίδας
  - Χειρισμός από το ΛΣ
- ◆ Δημιουργία διεργασιών σε συστήματα Εικονικής Μνήμης
  - Αντιγραφή κατά την Εγγραφή (Copy-on-Write)
- ◆ Απεικόνιση Αρχείων στην Εικονική Μνήμη

# Δυναμική σύνδεση με μοιραζόμενες βιβλιοθήκες

Max

Στοίβα



μοιραζόμενη  
βιβλιοθήκη

Σωρός

Δεδομένα

Κείμενο

0

Εικονική  
Μνήμη  $P_0$

Max

Στοίβα



μοιραζόμενη  
βιβλιοθήκη

Σωρός

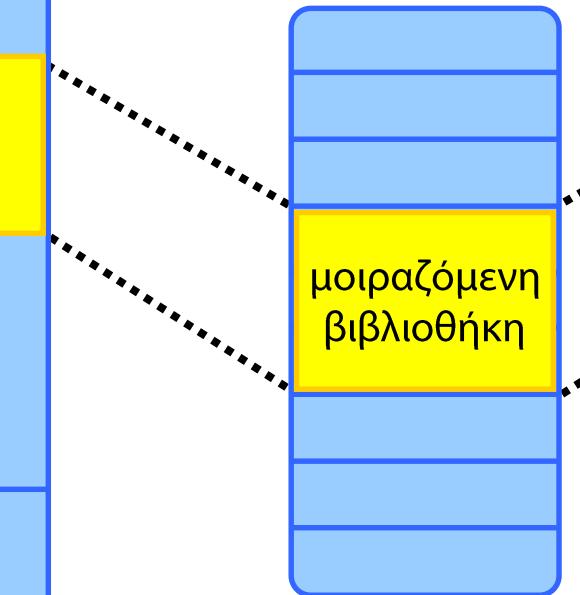
Δεδομένα

Κείμενο

0

Εικονική  
Μνήμη  $P_1$

Φυσική  
Μνήμη



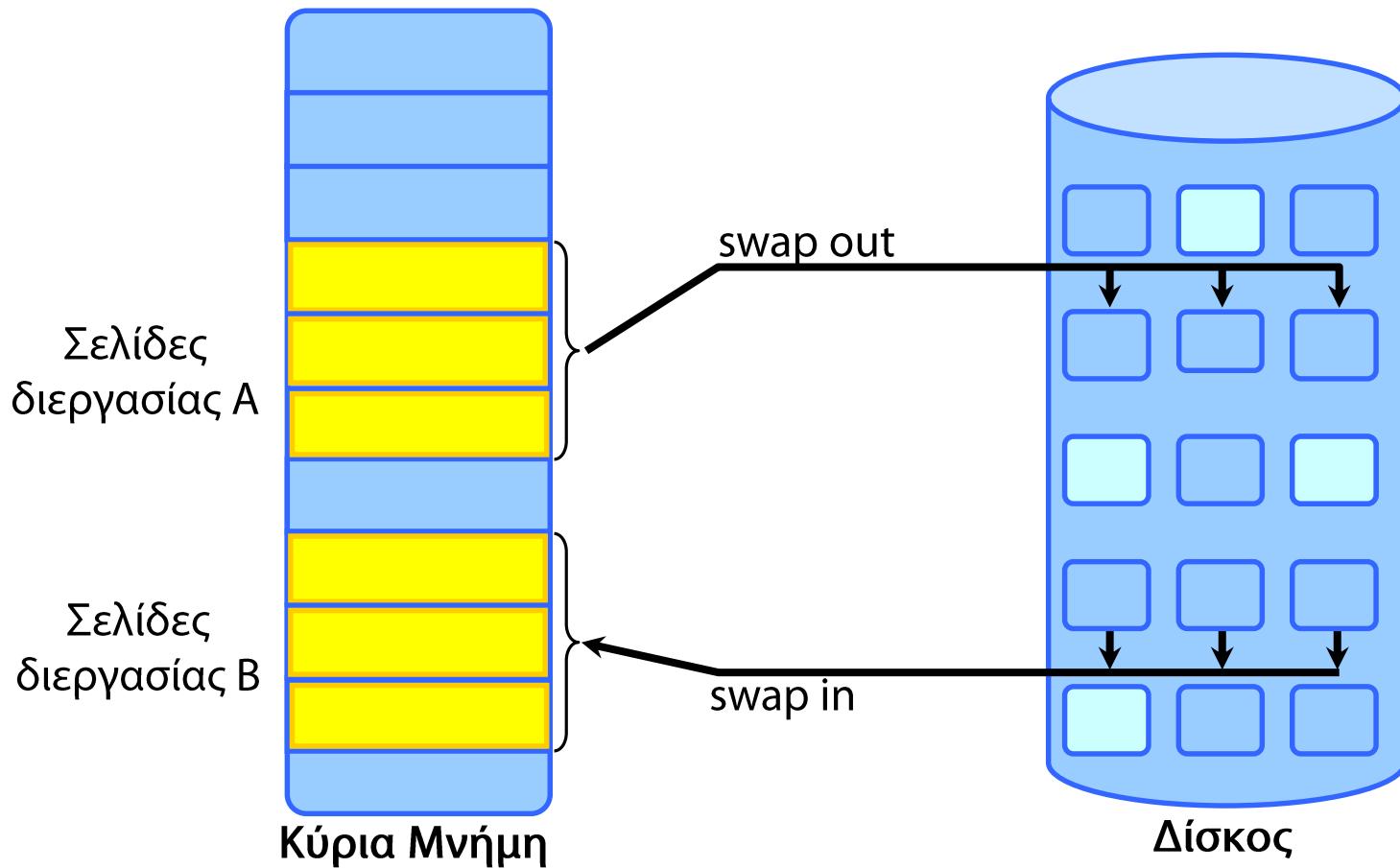
# Εικονική Μνήμη - Σύνοψη

- ◆ Σελιδοποίηση
- ◆ Εικονική Μνήμη με σελιδοποίηση
- ◆ Δυναμική σύνδεση με μοιραζόμενες βιβλιοθήκες
- ◆ **Σελιδοποίηση κατ' απαίτηση**
- ◆ Σφάλμα σελίδας
  - Χειρισμός από το ΛΣ
- ◆ Δημιουργία διεργασιών σε συστήματα Εικονικής Μνήμης
  - Αντιγραφή κατά την Εγγραφή (Copy-on-Write)
- ◆ Απεικόνιση Αρχείων στην Εικονική Μνήμη

# Σελιδοποίηση κατ' απαίτηση (Demand Paging)

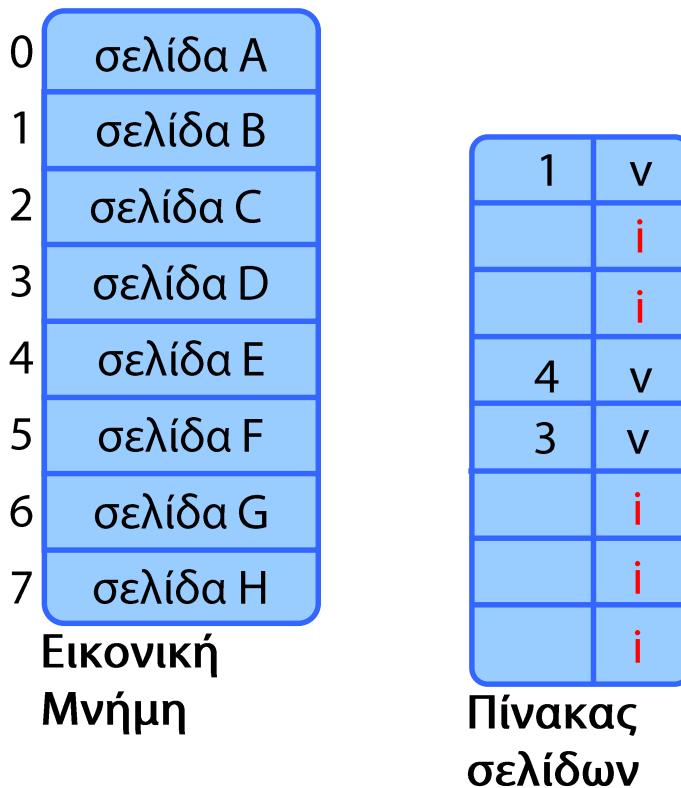
- ◆ Σελίδες εικονικής μνήμης ζουν σε **φυσική μνήμη** και **δίσκο**
- ◆ Όταν μια διεργασία χρειαστεί συγκεκριμένη σελίδα (αναφορά), **τότε** φέρε τη στη μνήμη
  - ➡ Πλεονεκτήματα:
    - Λιγότερη Είσοδος / Έξοδος
    - Γρηγορότερη απόκριση – η διεργασία ξεκινά αμέσως
    - Περισσότερες διεργασίες / περισσότεροι χρήστες
- ◆ Παρόμοια με την Εναλλαγή (swapping)
  - ➡ Σε επίπεδο σελίδας, με **page**

# Μεταφορά σελίδων από/προς Δίσκο



- ◆ 'Όχι κατ' ανάγκη σε συνεχόμενα πλαίσια ή συνεχόμενα τμήματα στο δίσκο

# Bit εγκυρότητας στον πίνακα σελίδων



- ◆ Είναι κάθε αναφορά στο χώρο εικονικών διευθύνσεων **έγκυρη**;
- ◆ Επιπλέον bit στον πίνακα σελίδων: **Valid – Invalid**
- ◆ Τι συμβαίνει αν γίνει αναφορά σε **Invalid** σελίδα; **Page Fault**

# Εικονική Μνήμη - Σύνοψη

- ◆ Σελιδοποίηση
- ◆ Εικονική Μνήμη με σελιδοποίηση
- ◆ Δυναμική σύνδεση με μοιραζόμενες βιβλιοθήκες
- ◆ Σελιδοποίηση κατ' απαίτηση
- ◆ **Σφάλμα σελίδας**
  - Χειρισμός από το ΛΣ
- ◆ Δημιουργία διεργασιών σε συστήματα Εικονικής Μνήμης
  - Αντιγραφή κατά την Εγγραφή (Copy-on-Write)
- ◆ Απεικόνιση Αρχείων στην Εικονική Μνήμη

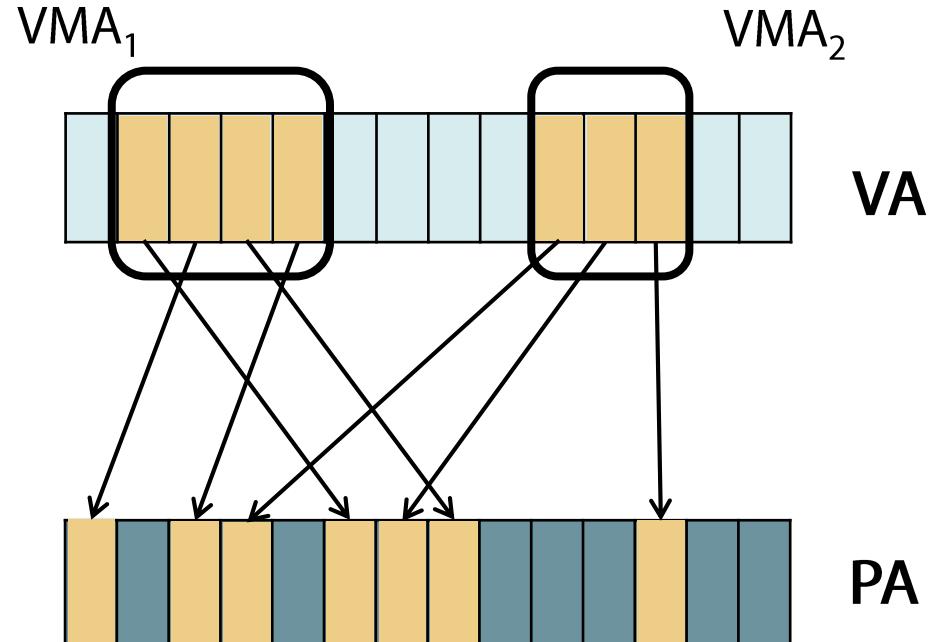
# Σφάλμα Σελίδας – Page Fault (1)

- ◆ **Εξαίρεση:** Το ΛΣ ξυπνάει
  - ➡ Το πρόγραμμα έκανε αναφορά σε άκυρη σελίδα
- ◆ **Γιατί;** Το ΛΣ μελετά το χάρτη μνήμης του
  - ➡ Μη επιτρεπόμενη πρόσβαση – π.χ. λόγω bug
    - τερματισμός προγράμματος (“Segmentation Fault”)
  - ➡ Επιτρεπόμενη πρόσβαση:
    - *η σελίδα είναι στο δίσκο*
      - πρέπει να έρθει στην Κύρια Μνήμη
      - έως τότε, η διεργασία σε τι κατάσταση βρίσκεται;
    - *η σελίδα δεν έχει ακόμα προσπελαστεί* (π.χ. έχει γίνει malloc αλλά όχι πρόσβαση / αρχικοποίηση)
      - πρέπει να ανατεθεί ένα πλαίσιο φυσικής μνήμης στη σελίδα

# 'Εγκυρο fault – demand paging

- Virtual Memory Areas (VMA) → malloc(), mmap(), etc.
- Φυσικά πλάισια → on demand δέσμευση (κατά τη πρόσβαση)

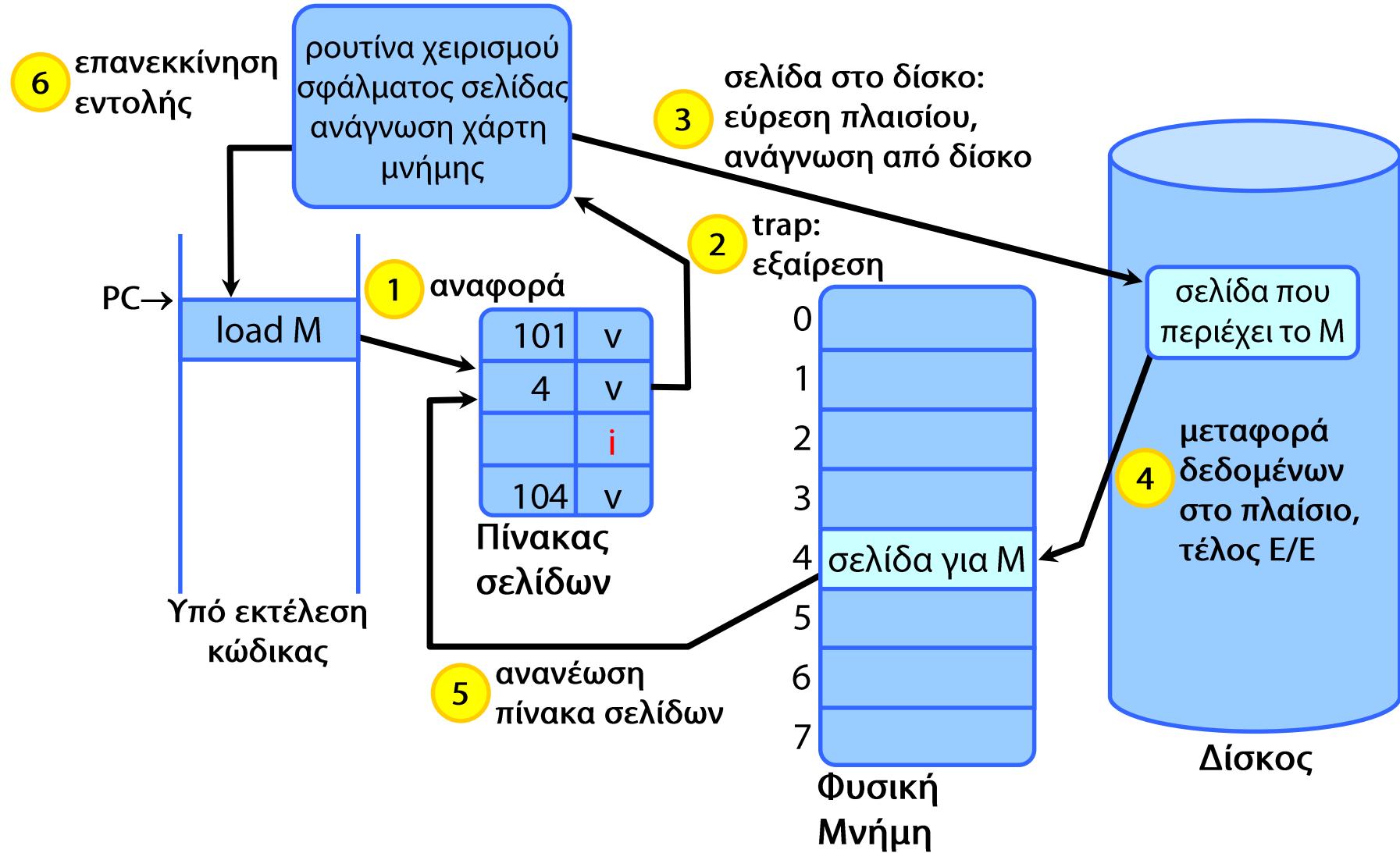
Page Fault



# Εξυπηρέτηση σφάλματος σελίδας (1)

- ◆ Βρες ένα ελεύθερο πλαίσιο μνήμης
  - ➡ Υπάρχει πάντα?
- ◆ Φέρε τα δεδομένα της σελίδας από το δίσκο σε αυτό το πλαίσιο (αν βρίσκονται στο δίσκο)
- ◆ Θέσε ανάλογη εγγραφή στον πίνακα σελίδων
  - ➡ Με bit εγκυρότητας **Valid**
- ◆ Επίστρεψε από τη διακοπή, στην εντολή που προκάλεσε την εξαίρεση
- ◆ Η ΚΜΕ τι κάνει, μέχρι να έρθει η σελίδα στην μνήμη;

# Εξυπηρέτηση σφάλματος σελίδας (2)



# Εξυπηρέτηση σφάλματος σελίδας (3)

- ◆ Πόσο κοστίζει να έρθει μια σελίδα από το δίσκο;
  - ➡ Ας πούμε ~10ms. Τι κάνει η ΚΜΕ όλη αυτή την ώρα;
- ◆ Χειρισμός σφάλματος σελίδας σε πραγματικά συστήματα
  - ➡ ασύγχρονος χειρισμός, σε **δύο στάδια**
- ◆ Αρχικοποίηση Ε/Ε
  - ➡ όταν ενεργοποιείται η ρουτίνα χειρισμού σφάλματος σελίδας
  - ➡ το ΛΣ ζητά από το δίσκο να γεμίσει ένα πλαίσιο μνήμης με τη ζητούμενη σελίδα (DMA)
  - ➡ η διεργασία αλλάζει κατάσταση, περνά **σε αναμονή**
- ◆ Ολοκλήρωση Ε/Ε
  - ➡ όταν ο δίσκος ολοκληρώσει τη μεταφορά → διακοπή
  - ➡ το ΛΣ διορθώνει τον πίνακα σελίδων
  - ➡ η διεργασία γίνεται **έτοιμη**

# Σελιδοποίηση κατ' απαίτηση – Κόστος

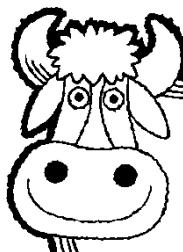
- ◆ Ρυθμός σφαλμάτων σελίδας  $p$ ,  $0 \leq p \leq 1$ 
  - ➡  $p = 0$ : δεν γίνονται ποτέ σφάλματα σελίδας
  - ➡  $p = 1$ : κάθε πρόσβαση προκαλεί σφάλμα σελίδας
- ◆ Πραγματικός χρόνος πρόσβασης
  - ➡ Effective Access Time
    - $EAT = (1 - p) \times t_m + p \times (\text{κόστος σφάλματος σελίδας} + \text{κόστος page in} + \text{κόστος επανεκκίνησης})$
  - ➡ π.χ., για  $p = 1\%$ ,  $t_m = 200\text{ns}$ ,  $t_{pf} = 8\text{ms} = 8 \times 10^6\text{ns}$ 
    - $EAT = 8200\text{ns}$ , χρόνος πρόσβασης 40 φορές μεγαλύτερος!

# Εικονική Μνήμη - Σύνοψη

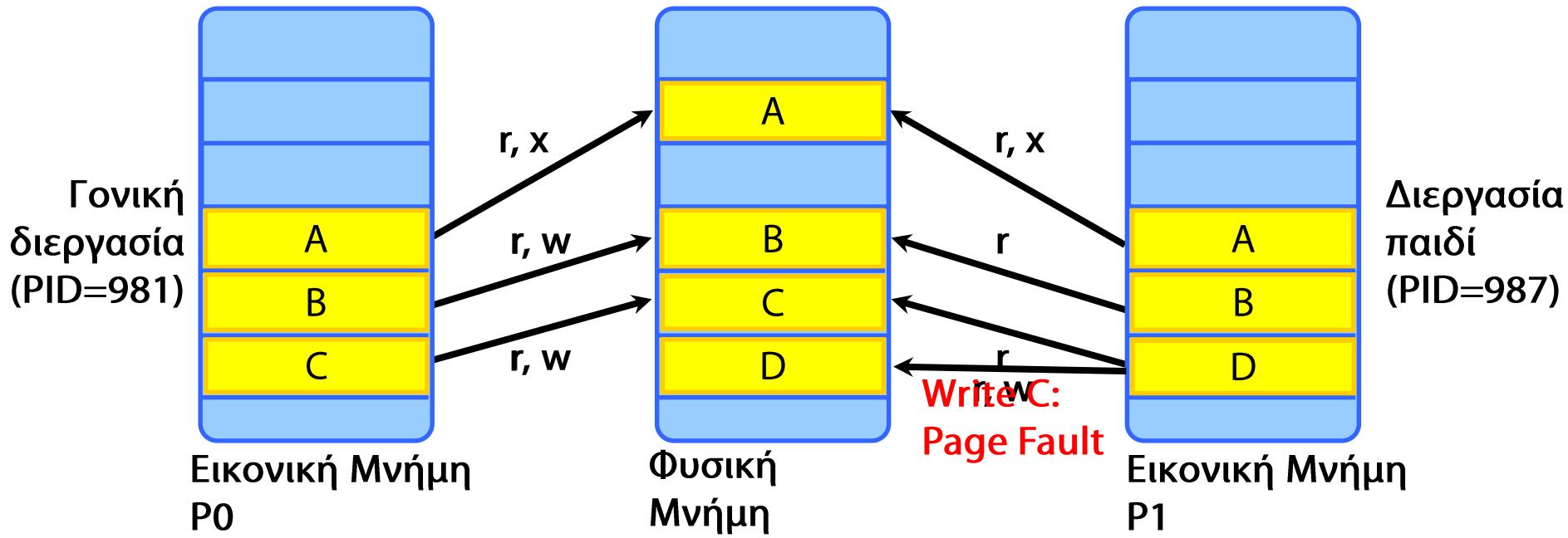
- ◆ Σελιδοποίηση
- ◆ Εικονική Μνήμη με σελιδοποίηση
- ◆ Δυναμική σύνδεση με μοιραζόμενες βιβλιοθήκες
- ◆ Σελιδοποίηση κατ' απαίτηση
- ◆ Σφάλμα σελίδας
  - Χειρισμός από το ΛΣ
- ◆ **Δημιουργία διεργασιών σε συστήματα Εικονικής Μνήμης**
  - Αντιγραφή κατά την Εγγραφή (Copy-on-Write)
- ◆ Απεικόνιση Αρχείων στην Εικονική Μνήμη

# Δημιουργία νέας διεργασίας – π.χ. fork()

- ◆ Η νέα διεργασία είναι **αντίγραφο** της παλιάς
- ◆ Κληρονομεί όλα τα ανοιχτά αρχεία και αντίγραφο της μνήμης της αρχικής διεργασίας
- ◆ Από εκεί και πέρα, εκτελείται ανεξάρτητα
- ◆ Πόσο κοστίζει η δημιουργία της;
  - ➡ Αντιγράφεται όλη η μνήμη της παλιάς διεργασίας
  - ➡ Πάντα;
- ◆ **'Οχι!** Σε συστήματα εικονικής μνήμης
  - ➡ Αντιγραφή κατά την Εγγραφή – Copy-on-Write (COW)



# Copy-on-Write



- ◆ Μετά τη δημιουργία νέας διεργασίας, *μοιράζονται* σελίδες
  - με διαφορετικά δικαιώματα πρόσβασης
- ◆ (Ότ)αν η διεργασία προσπαθήσει να γράψει σε κάποια σελίδα
  - Εύρεση νέου πλαισίου, αντιγραφή και αλλαγή του πίνακα σελίδων

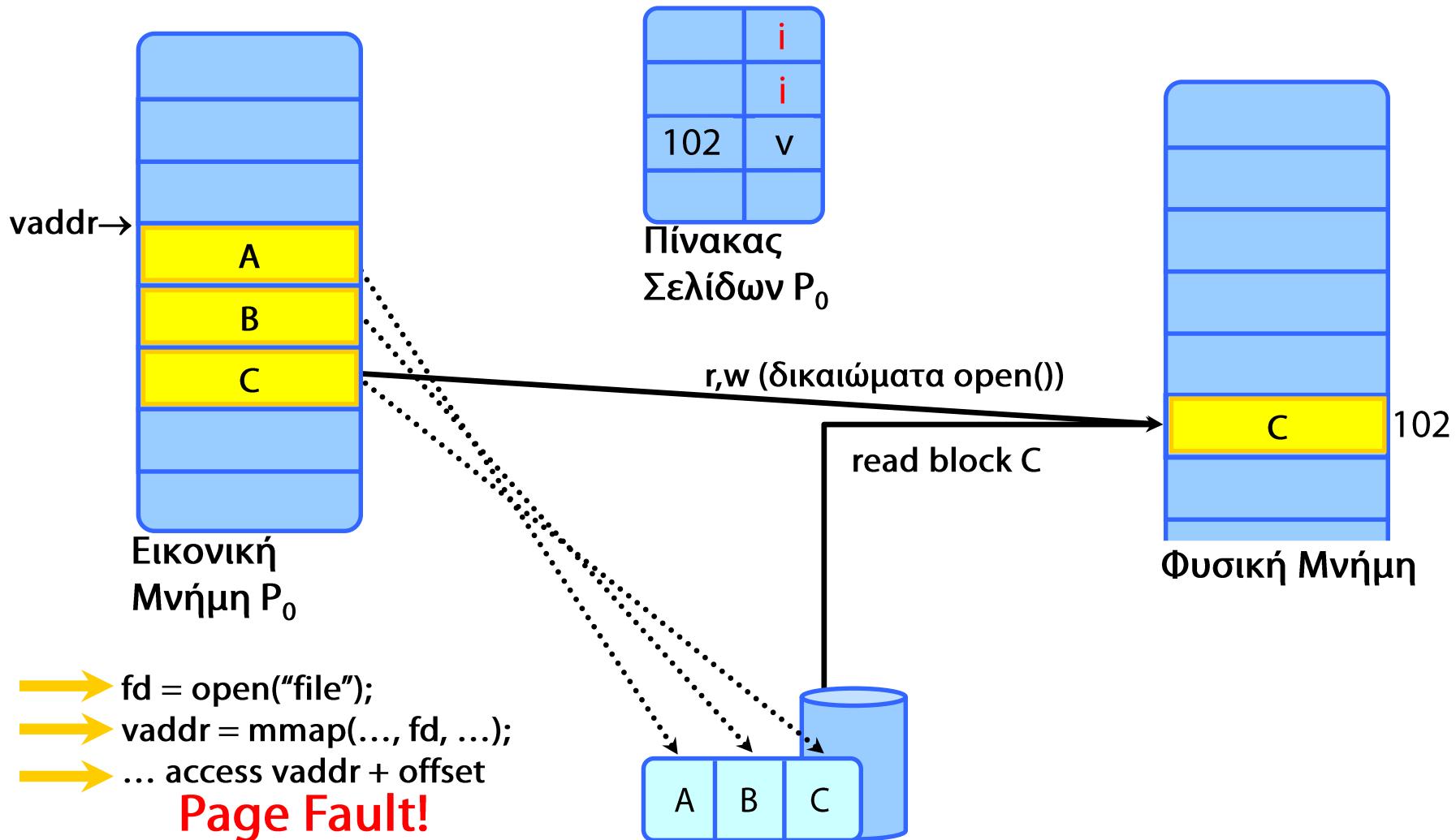
# Εικονική Μνήμη - Σύνοψη

- ◆ Σελιδοποίηση
  - Λογικές διευθύνσεις, Πίνακες Σελίδων
- ◆ Εικονική Μνήμη με σελιδοποίηση
- ◆ Δυναμική σύνδεση με μοιραζόμενες βιβλιοθήκες
- ◆ Σελιδοποίηση κατ' απαίτηση
- ◆ Σφάλμα σελίδας
  - Χειρισμός από το ΛΣ
- ◆ Δημιουργία διεργασιών σε συστήματα Εικονικής Μνήμης
  - Αντιγραφή κατά την Εγγραφή (Copy-on-Write)
- ◆ **Απεικόνιση Αρχείων στην Εικονική Μνήμη**

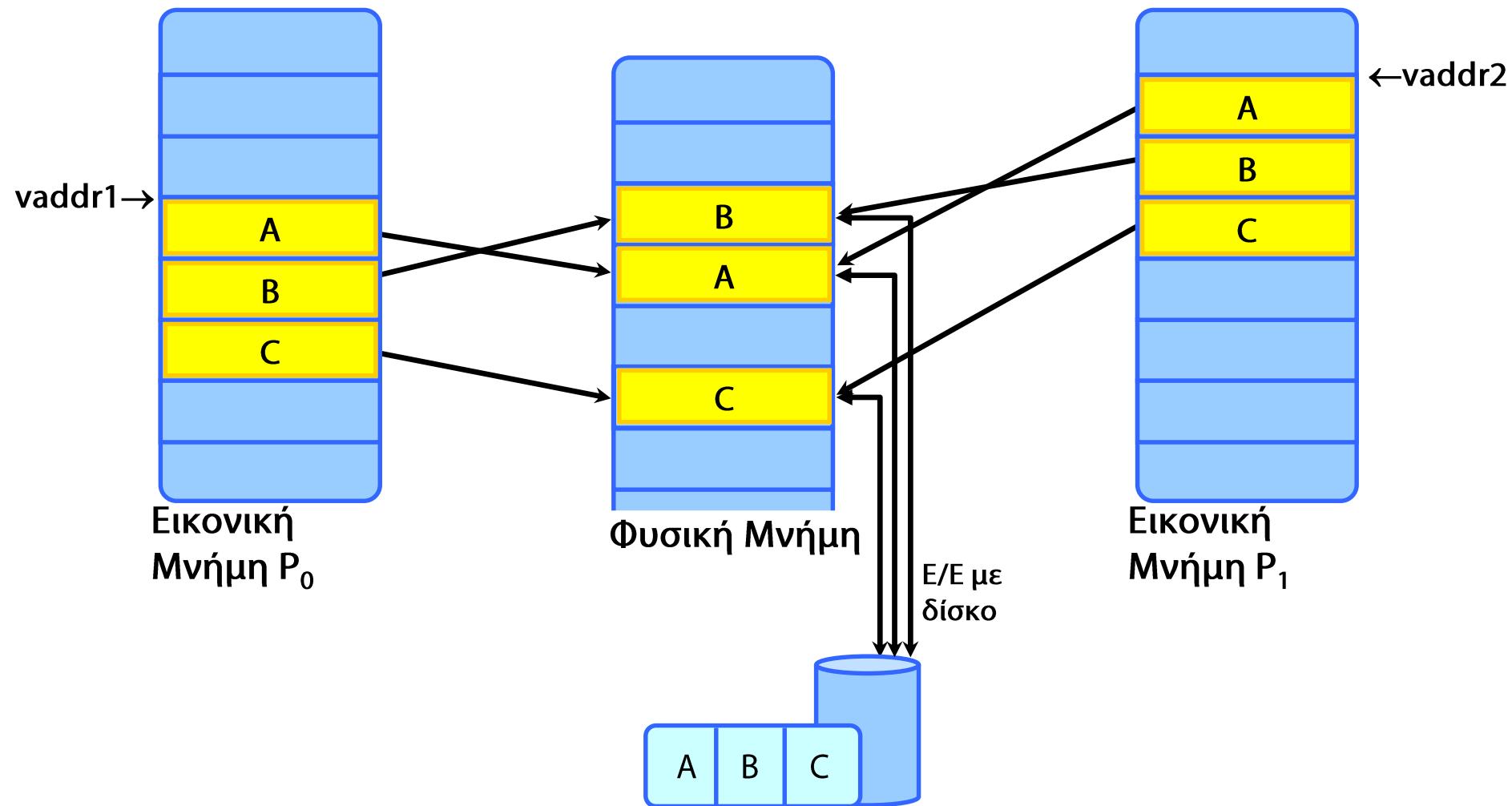
# Απεικόνιση Αρχείων στη Μνήμη (1)

- ◆ Αντί για `read()` και `write()`, πρόσβαση σε αρχεία με **load** και **store**
  - απλουστεύει τον προγραμματισμό
- ◆ το αρχείο **απεικονίζεται** σε συνεχόμενο χώρο εικονικής μνήμης
- ◆ Ε/Ε με σελιδοποίηση κατ'απαίτηση
  - **όταν** η διεργασία αναφερθεί σε μια νέα σελίδα
  - το ΛΣ γεμίζει το αντίστοιχο πλαίσιο με δεδομένα από το δίσκο

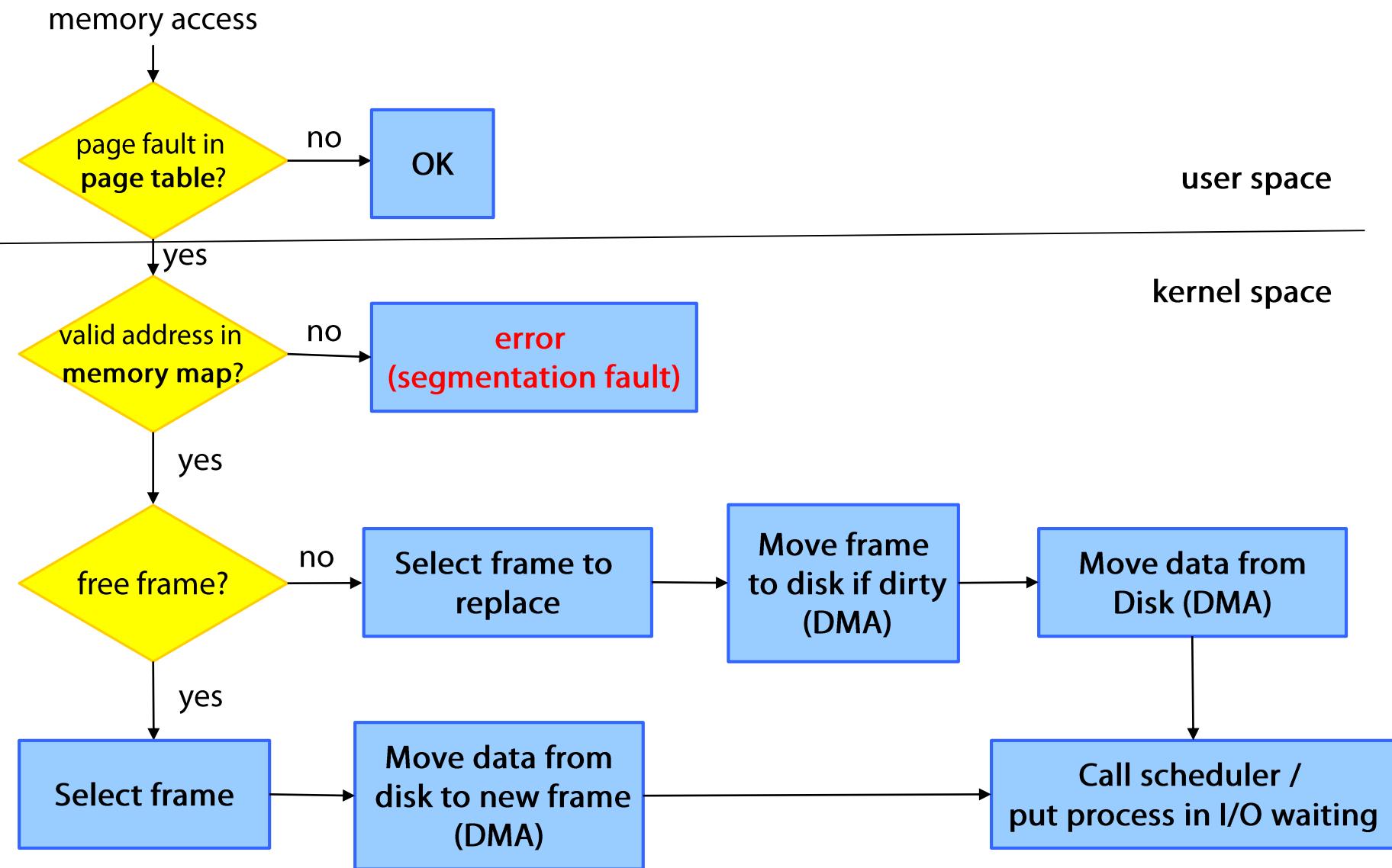
# Απεικόνιση Αρχείων στη Μνήμη (2)



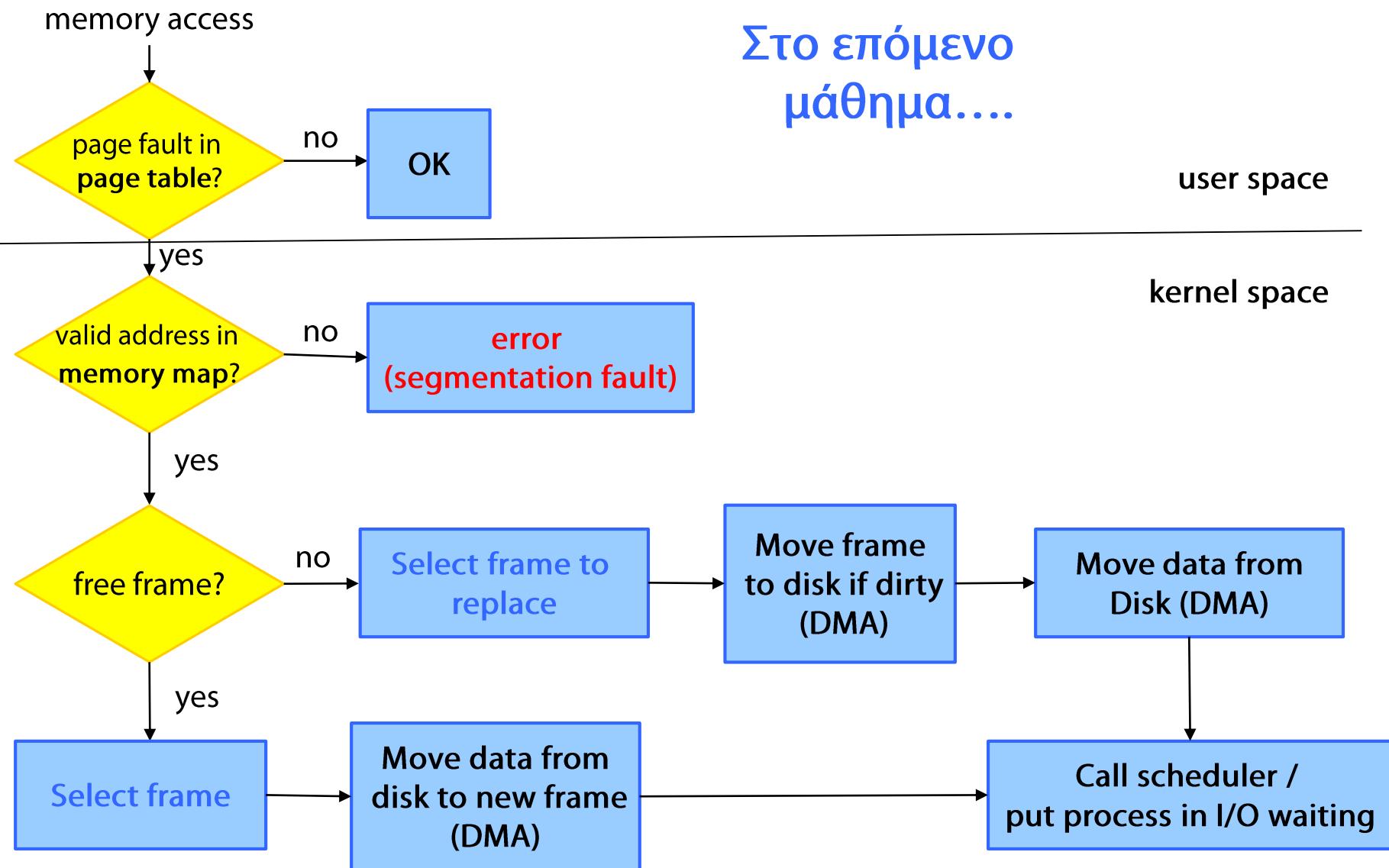
# Απεικόνιση Αρχείων στη Μνήμη (2)



# Εικονική Μνήμη με Σελιδοποίηση: Η συνολική εικόνα



# Εικονική Μνήμη με Σελιδοποίηση: Η συνολική εικόνα



# Ερωτήσεις;



και στη λίστα:

[OS@lists.cslab.ece.ntua.gr](mailto:OS@lists.cslab.ece.ntua.gr)