

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

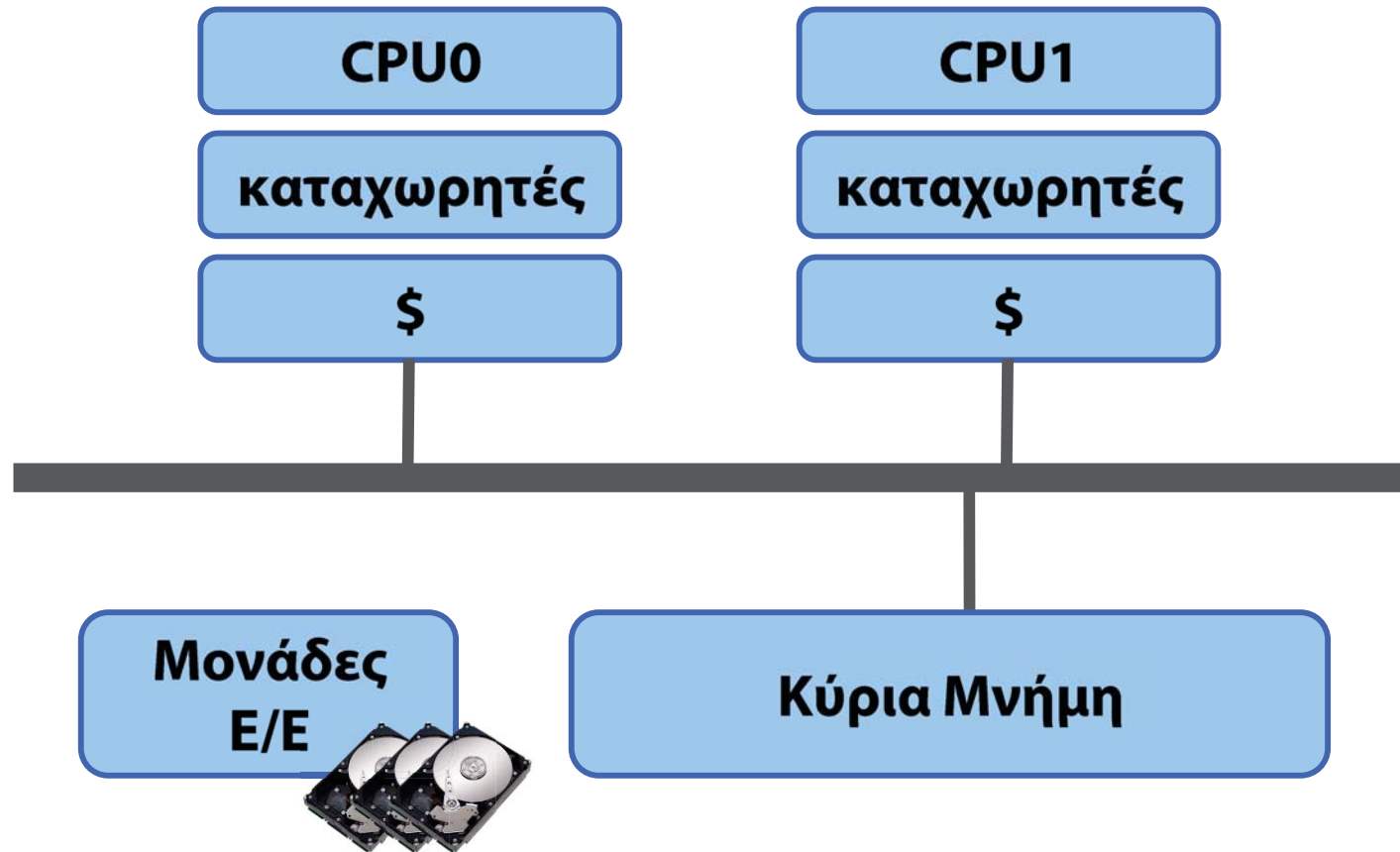
Εικονική Μνήμη - Σύνοψη

◆ Σελιδοποίηση

➔ Λογικές διευθύνσεις, Πίνακες Σελίδων

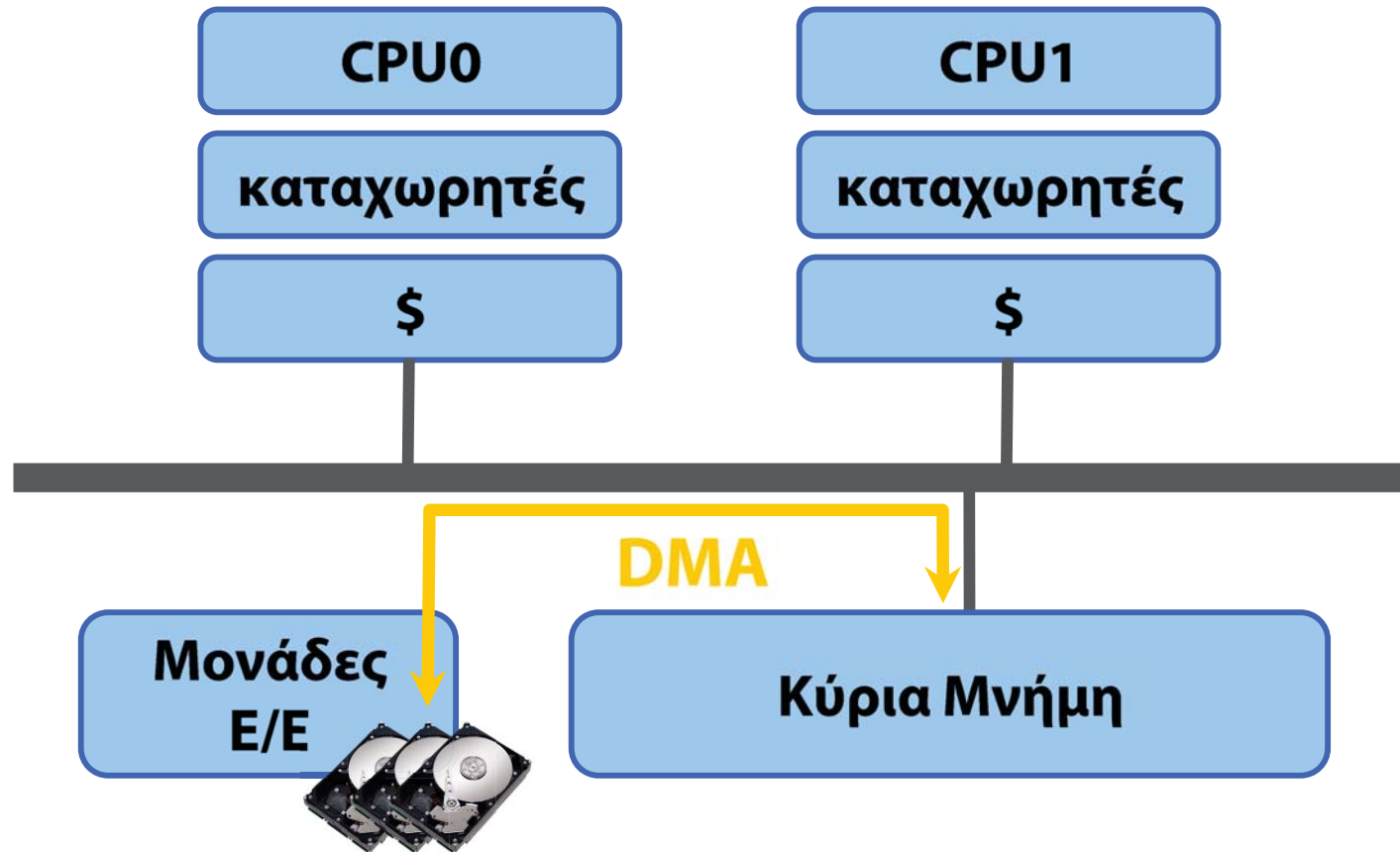
- ◆ Εικονική Μνήμη με σελιδοποίηση
- ◆ Δυναμική σύνδεση με μοιραζόμενες βιβλιοθήκες
- ◆ Σελιδοποίηση κατ' απαίτηση
- ◆ Σφάλμα σελίδας
 - ➔ Χειρισμός από το ΛΣ
- ◆ Δημιουργία διεργασιών σε συστήματα Εικονικής Μνήμης
 - ➔ Αντιγραφή κατά την Εγγραφή (Copy-on-Write)
- ◆ Απεικόνιση Αρχείων στην Εικονική Μνήμη

Κύρια Μνήμη (1)



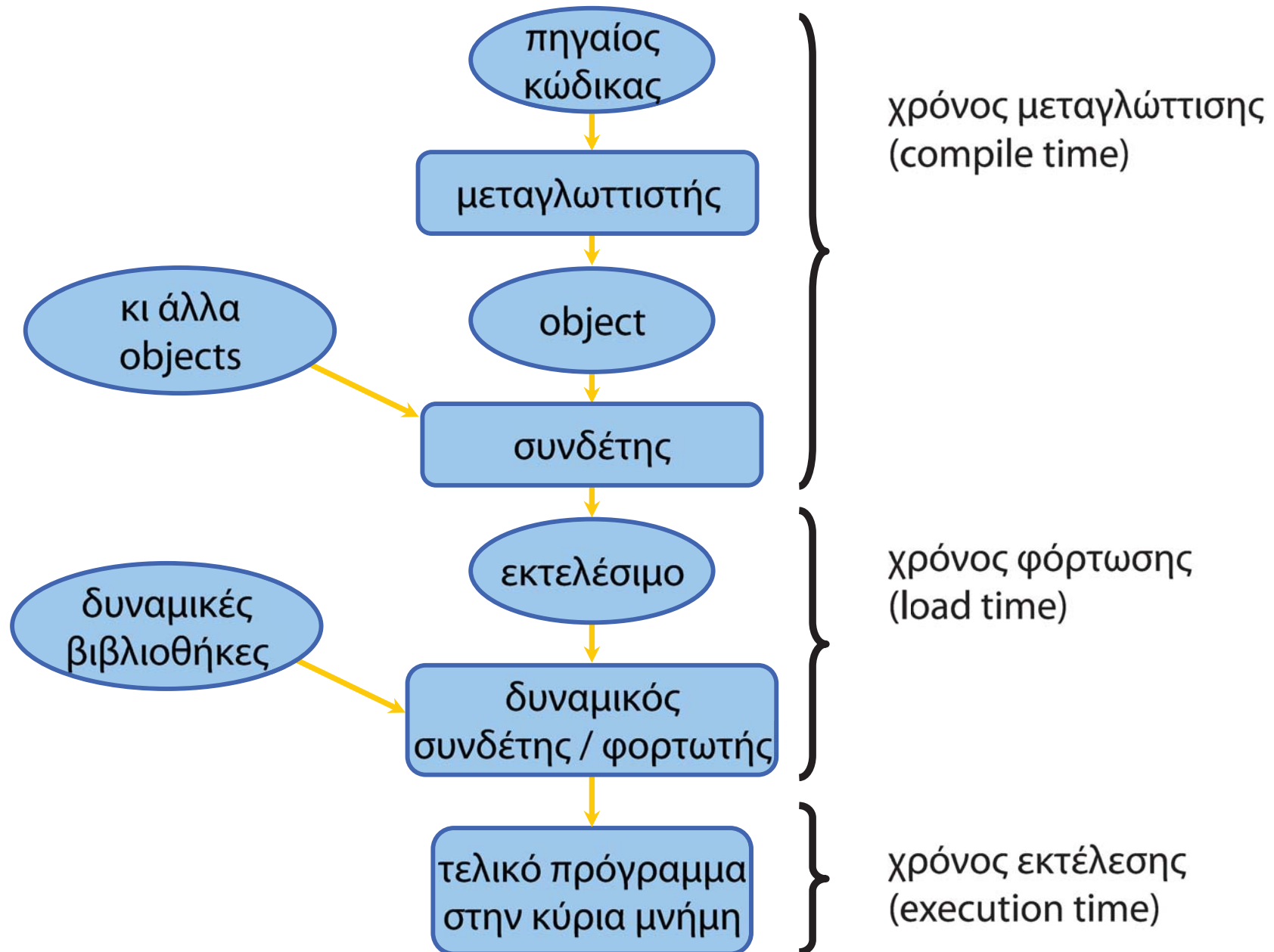
- ◆ Κάθε CPU αναφέρεται απευθείας σε καταχωρητές και μνήμη
- ◆ Συσκευές E/E εκτελούν Απευθείας Πρόσβαση στη Μνήμη (Direct Memory Access - DMA)

Κύρια Μνήμη (1)



- ◆ Κάθε CPU αναφέρεται απευθείας σε καταχωρητές και μνήμη
- ◆ Συσκευές Ε/Ε εκτελούν Απευθείας Πρόσβαση στη Μνήμη (Direct Memory Access - DMA)

Μεταγλώττιση – Φόρτωση - Εκτέλεση



Μετατοπίσιμος κώδικας

Μετατοπίσιμος κώδικας (πχ. object)

func1
movl \$0x5, val
call func2
ret

func2
... func2...

val
... 4 bytes...

func+2 ← &val
func+8 ← &func2

Πίνακας Μετατόπισης

Απόλυτος κώδικας

func ≡ **64**
70
76
ret

func2 ≡ **5000**
... func2...

val ≡ **7010**
... 4 bytes...

Συνδέτης / φορτωτής



- ◆ Ο κώδικας διορθώνεται με βάση τον πίνακα μετατόπισης
- ◆ Στο χρόνο μεταγλώττισης (συνδέτης) ή εκτέλεσης (φορτωτής)

Μετατοπίσιμος κώδικας

Μετατοπίσιμος κώδικας
(πχ. object)

func1
movl \$0x5, val
call func2
ret

func2
... func2...

val
... 4 bytes...

func+2 ← &val
func+8 ← &func2

Πίνακας Μετατόπισης

Συνδέτης / φορτωτής



Απόλυτος κώδικας

func ≡ **64**
70
76
ret

func2 ≡ **5000**
... func2...

val ≡ **7010**
... 4 bytes...

- ◆ Ο κώδικας διορθώνεται με βάση τον πίνακα μετατόπισης
- ◆ Στο χρόνο μεταγλώττισης (συνδέτης) ή εκτέλεσης (φορτωτής)

Μετατοπίσιμος κώδικας

Μετατοπίσιμος κώδικας (πχ. object)

func1
movl \$0x5, **val**
call **func2**
ret

func2
... func2...

val
... 4 bytes...

func+2 ← &val
func+8 ← &func2

Πίνακας Μετατόπισης

Συνδέτης / φορτωτής



Απόλυτος κώδικας

func ≡ **64**
70
76
ret
movl \$0x5, **7010**
call **\$5000**

func2 ≡ **5000**
... func2...

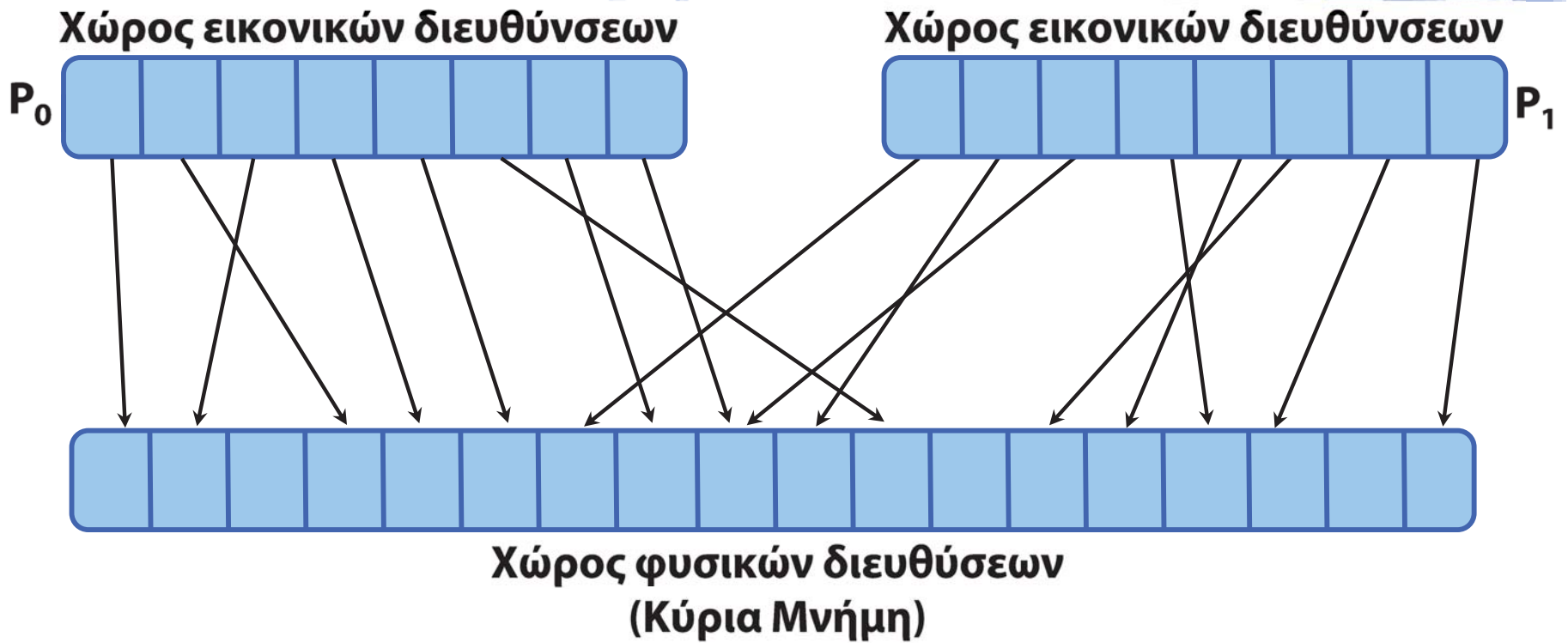
val ≡ **7010**
... 4 bytes...

- ◆ Ο κώδικας διορθώνεται με βάση τον πίνακα μετατόπισης
- ◆ Στο χρόνο μεταγλώττισης (συνδέτης) ή εκτέλεσης (φορτωτής)

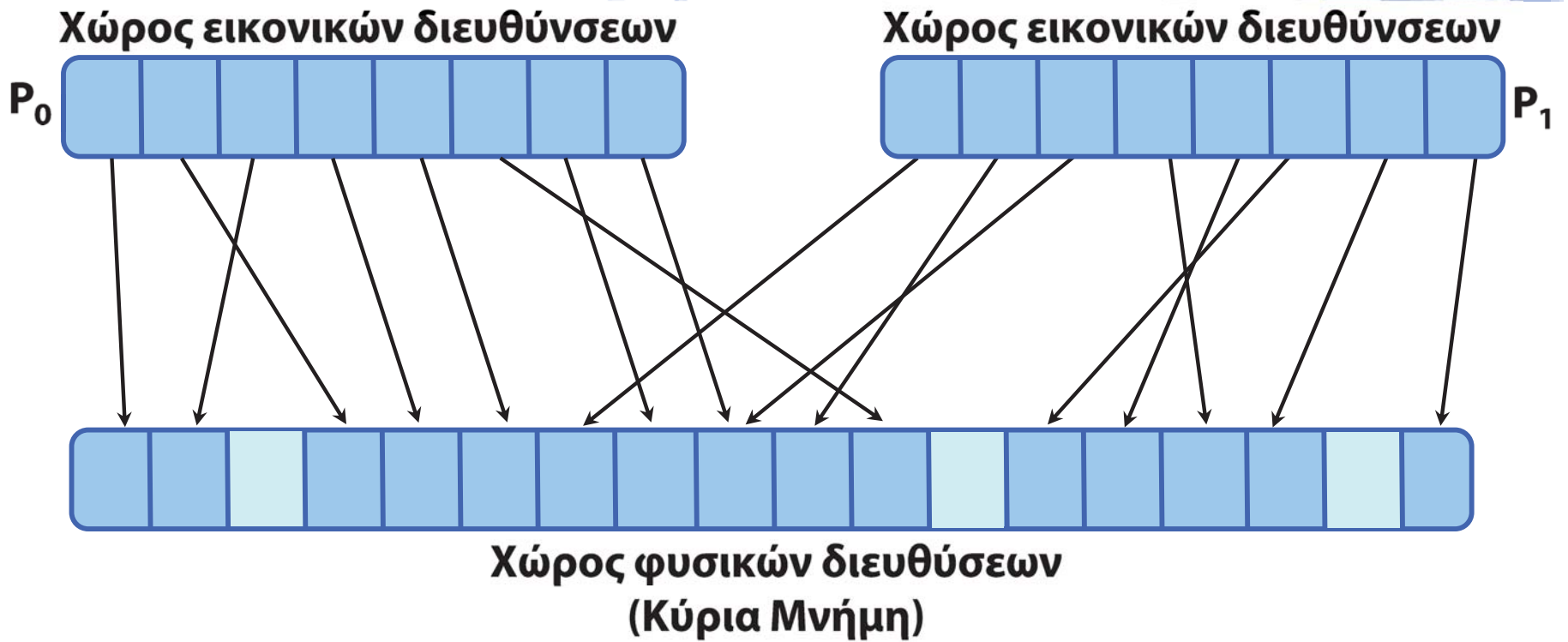
Σελιδοποίηση (1)

- ◆ Ο πιο συχνός τρόπος μετάφρασης διευθύνσεων
- ◆ 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

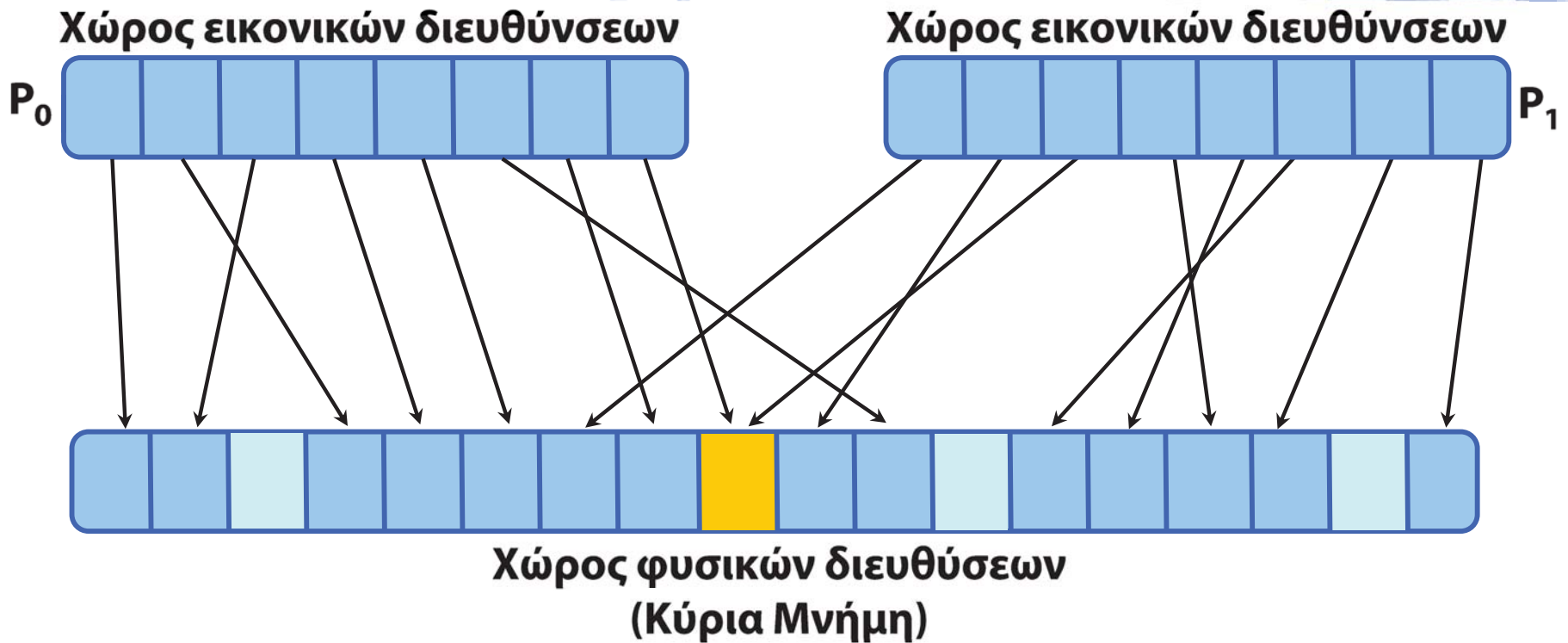
Σελιδοποίηση (2)



Σελιδοποίηση (2)

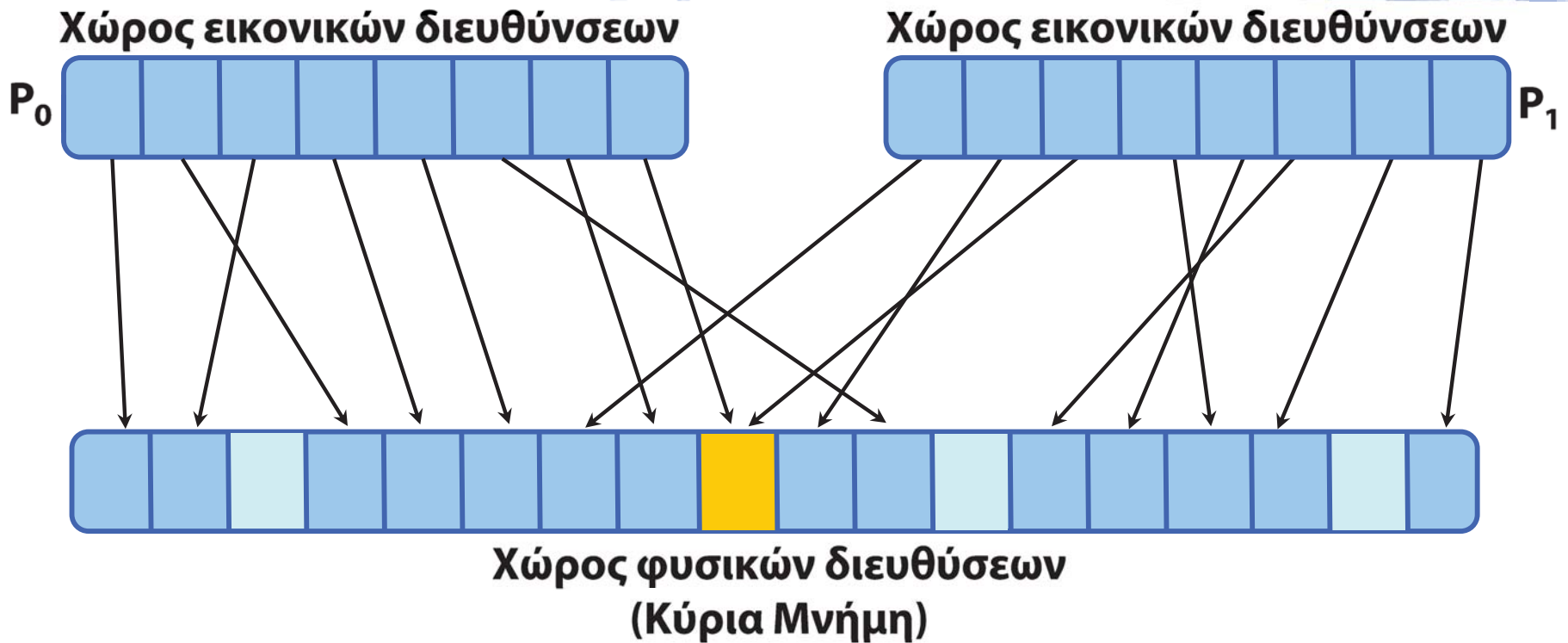


Σελιδοποίηση (2)



- ◆ 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 (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

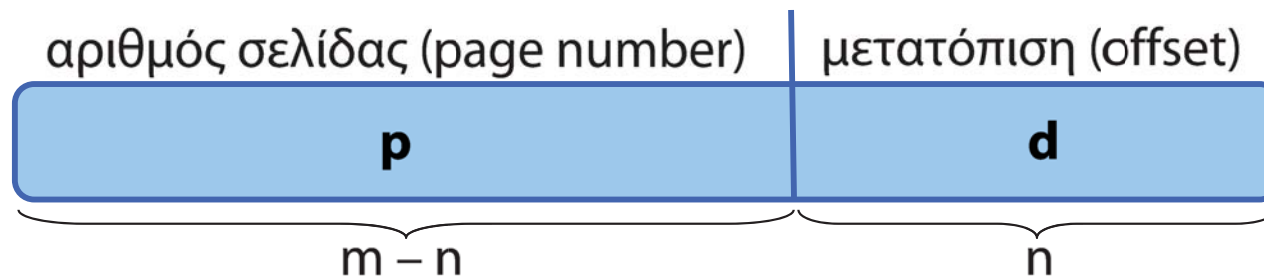
Σελιδοποίηση (3)



- ◆ 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
 - εντονότερος εσωτερικός κατακερματισμός
 - ➔ Μικρό μέγεθος
 - αποφεύγεται ο εσωτερικός κατακερματισμός
 - μεγαλύτερο κόστος διαχείρισης

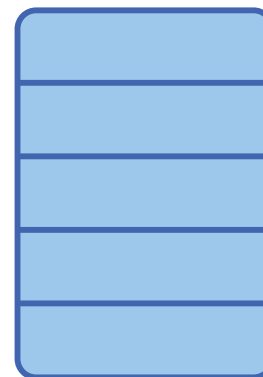
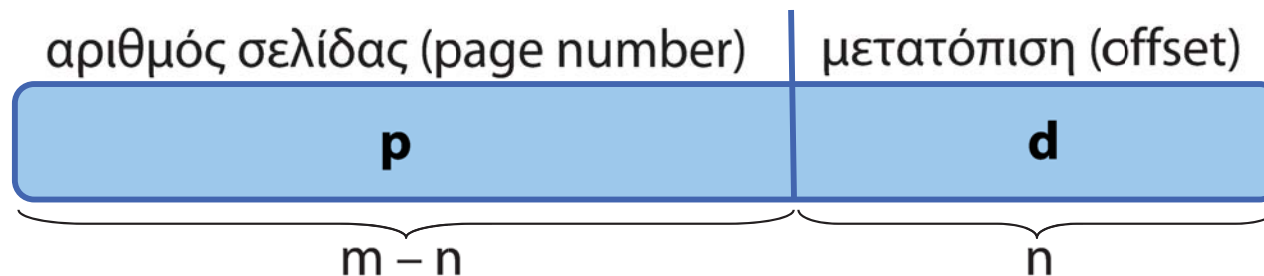
Σελιδοποίηση - Μετάφραση διευθύνσεων

- ◆ Μέγεθος σελίδας 2^n . **Γιατί** πρέπει να είναι δύναμη του 2;
- ◆ Διεύθυνση των m bits, χώρος εικονικών διευθύνσεων 2^m
- ◆ *Εικονική διεύθυνση*: αριθμός σελίδας + μετατόπιση



Σελιδοποίηση - Μετάφραση διευθύνσεων

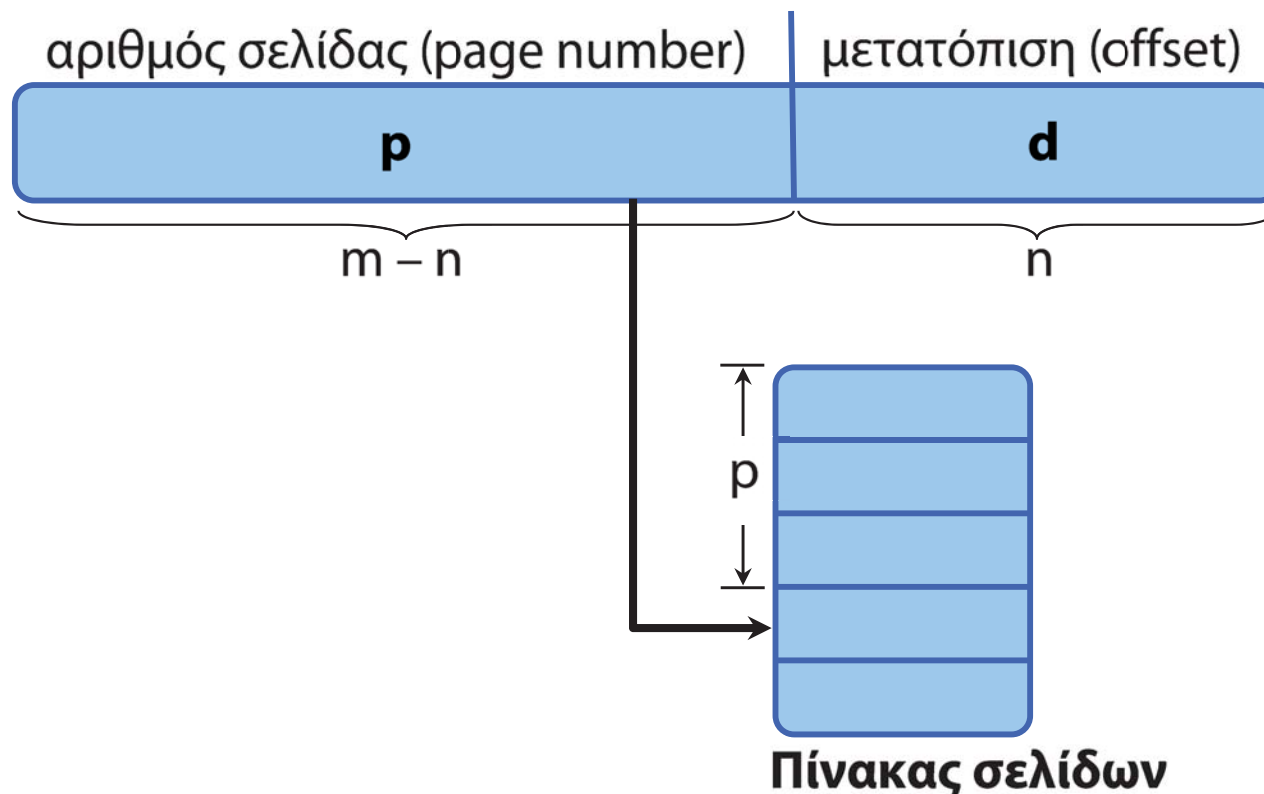
- ◆ Μέγεθος σελίδας 2^n . **Γιατί** πρέπει να είναι δύναμη του 2;
- ◆ Διεύθυνση των m bits, χώρος εικονικών διευθύνσεων 2^m
- ◆ *Εικονική διεύθυνση*: αριθμός σελίδας + μετατόπιση



Πίνακας σελίδων

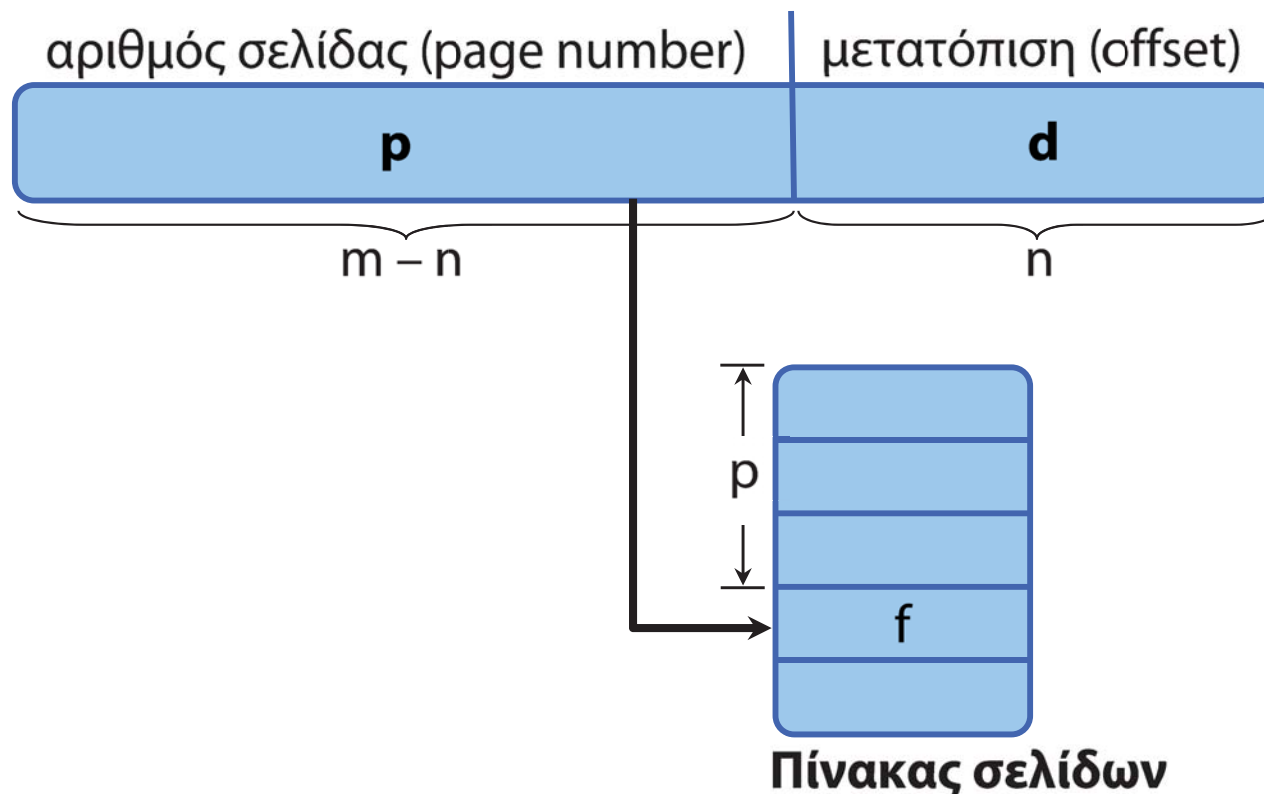
Σελιδοποίηση - Μετάφραση διευθύνσεων

- ◆ Μέγεθος σελίδας 2^n . **Γιατί** πρέπει να είναι δύναμη του 2;
- ◆ Διεύθυνση των m bits, χώρος εικονικών διευθύνσεων 2^m
- ◆ *Εικονική διεύθυνση*: αριθμός σελίδας + μετατόπιση

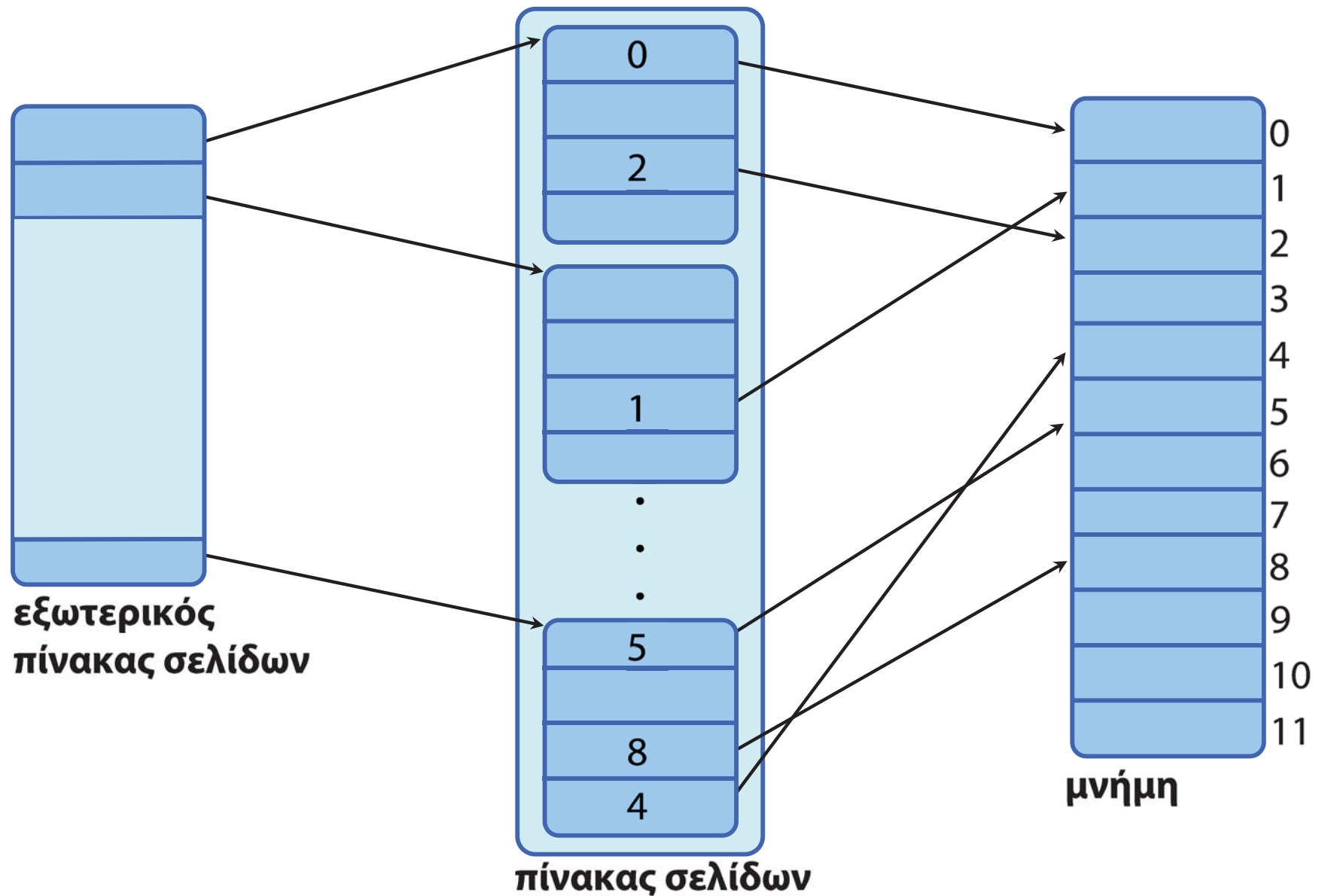


Σελιδοποίηση - Μετάφραση διευθύνσεων

- ◆ Μέγεθος σελίδας 2^n . **Γιατί** πρέπει να είναι δύναμη του 2;
- ◆ Διεύθυνση των m bits, χώρος εικονικών διευθύνσεων 2^m
- ◆ *Εικονική διεύθυνση*: αριθμός σελίδας + μετατόπιση



Διεπίπεδη Σελιδοποίηση



Σελιδοποίηση – Προστασία (1)



Σελιδοποίηση – Προστασία (1)

- ◆ Επιβολή δικαιωμάτων πρόσβασης
 - ➔ με bits προστασίας ανά σελίδα εικονικής μνήμης
- ◆ Π.χ., για μοιραζόμενη μνήμη
 - ➔ Μία διεργασία μπορεί να γράψει, οι άλλες μόνο διαβάζουν
- ◆ Bits πρόσβασης
 - ➔ **R**ead, **W**rite, **eX**ecute
- ◆ Bit εγκυρότητας
 - ➔ **V**alid, **I**nvalid
- ◆ 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
5
1
7
0

**Πίνακας
σελίδων**

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

**Φυσική
Μνήμη**

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**

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;

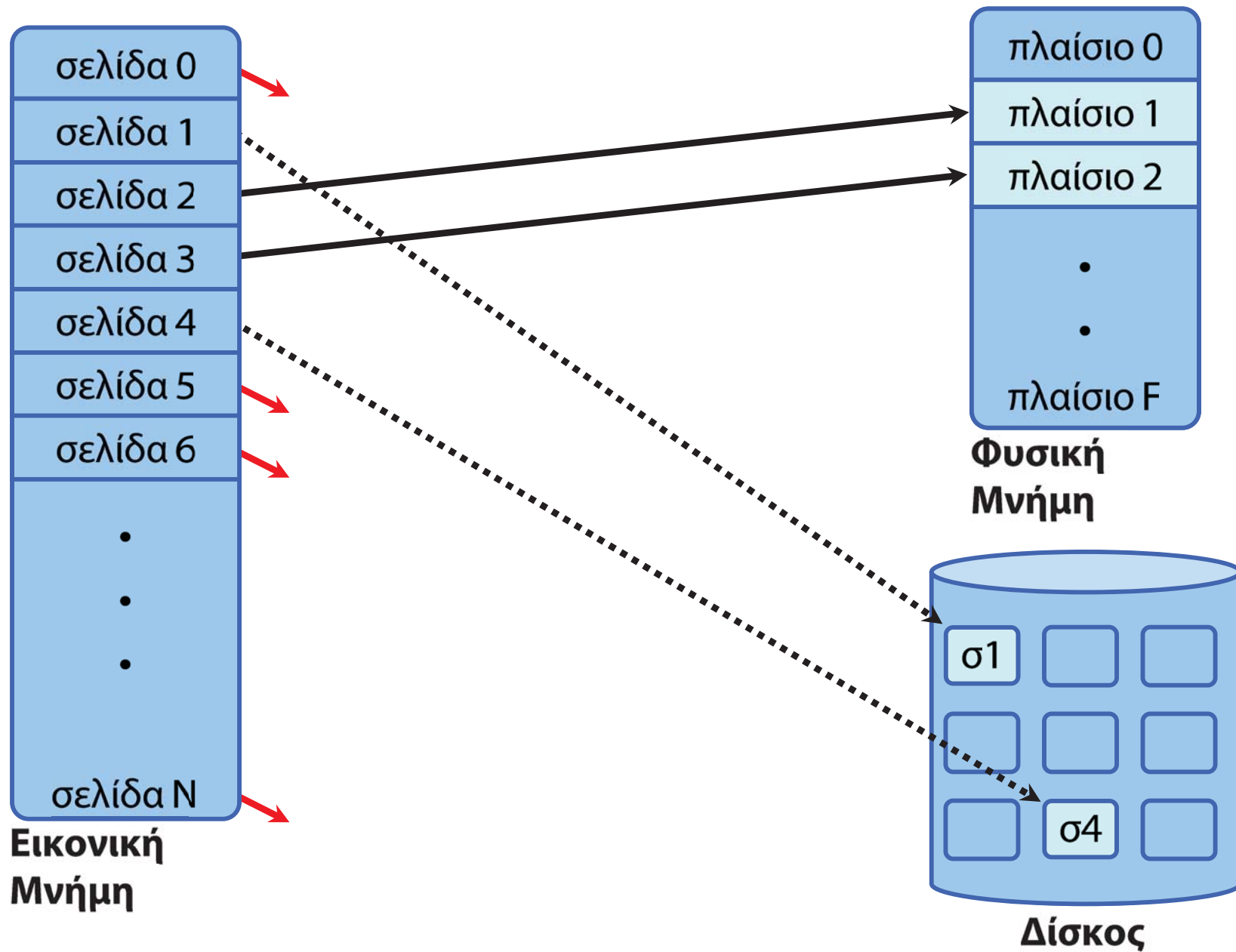
Εικονική Μνήμη - Σύνοψη

- ◆ Σελιδοποίηση
 - ➔ 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

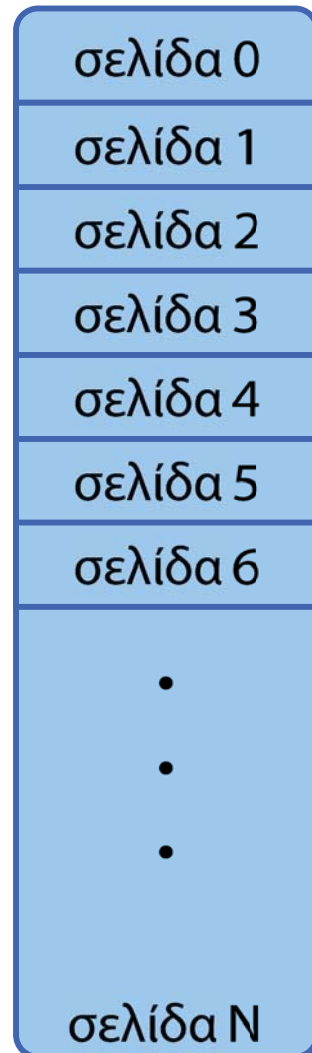
Virtual Memory – Εικονική Μνήμη

- ◆ **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 πού βρίσκονται;
- ◆ **Υλοποίηση:**
 - ➔ Κατ' απαίτηση σελιδοποίηση (demand paging)
 - ➔ Κατ' απαίτηση κατάτμηση (demand segmentation)

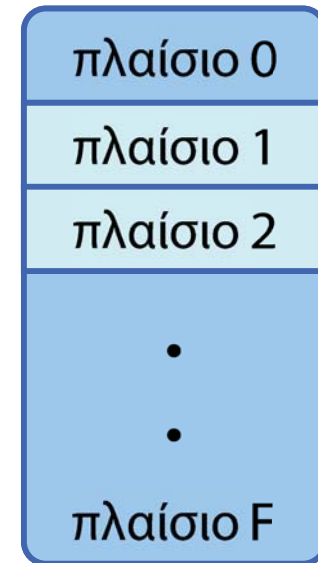
Περισσότερη Μνήμη!



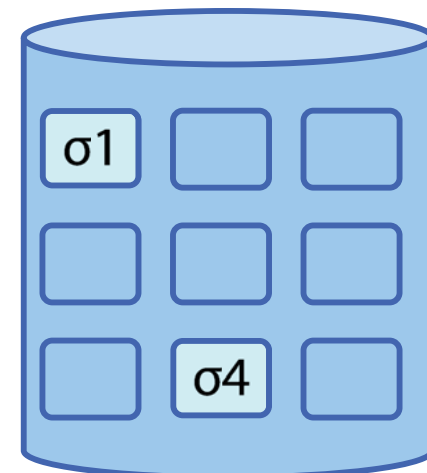
Περισσότερη Μνήμη!



**Εικονική
Μνήμη**

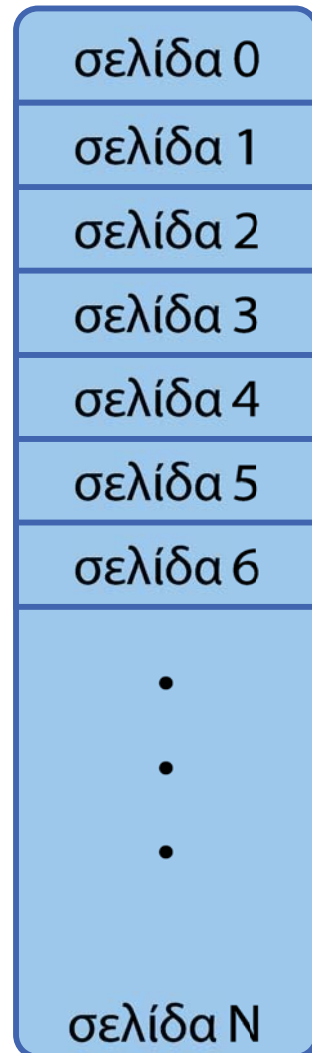


**Φυσική
Μνήμη**



Δίσκος

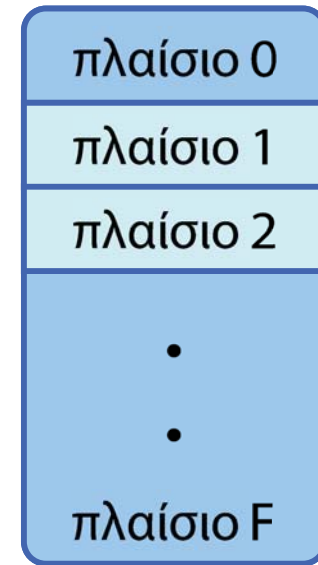
Περισσότερη Μνήμη!



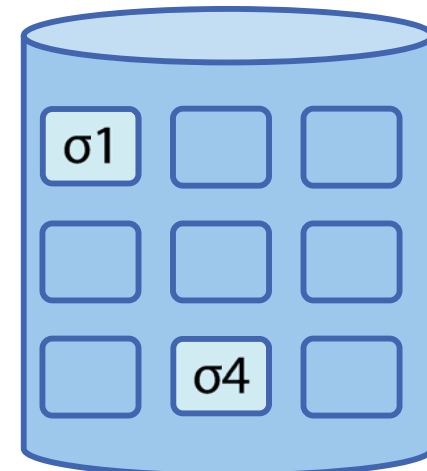
**Εικονική
Μνήμη**



**Πίνακας
Σελίδων
(αρκεί;)**

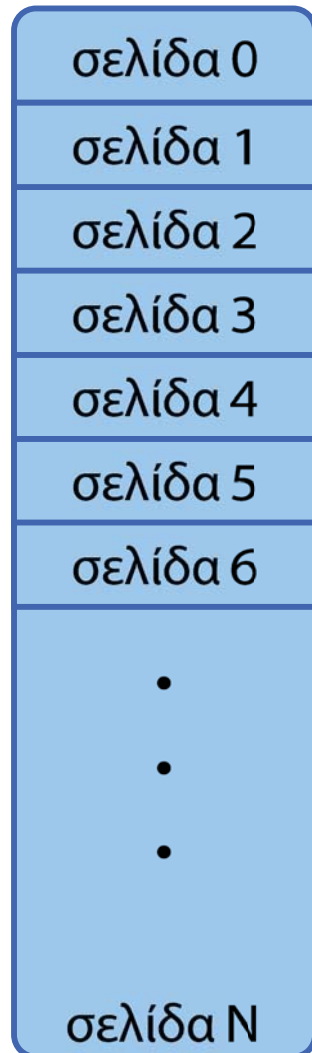


**Φυσική
Μνήμη**

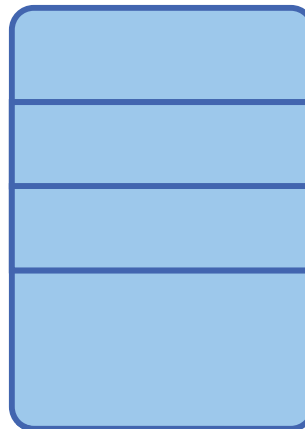


Δίσκος

Περισσότερη Μνήμη!



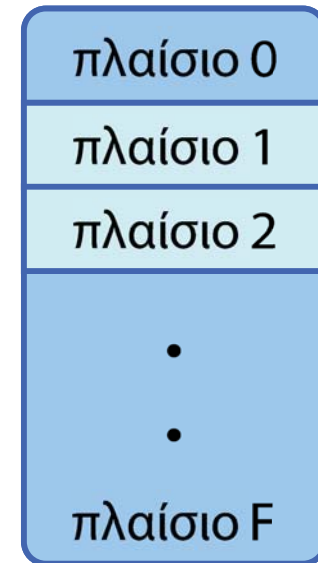
**Εικονική
Μνήμη**



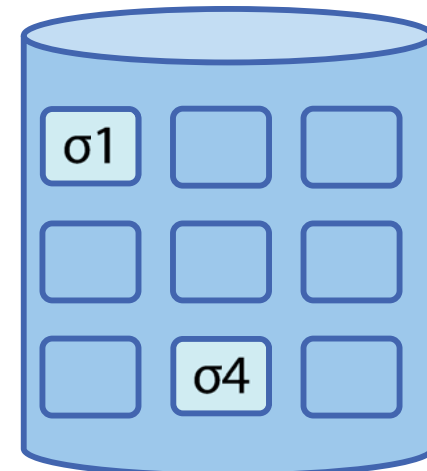
**Πίνακας
Σελίδων
(αρκεί;)**



**Χάρτης Μνήμης
(δομή του $\Lambda\Sigma$)**



**Φυσική
Μνήμη**

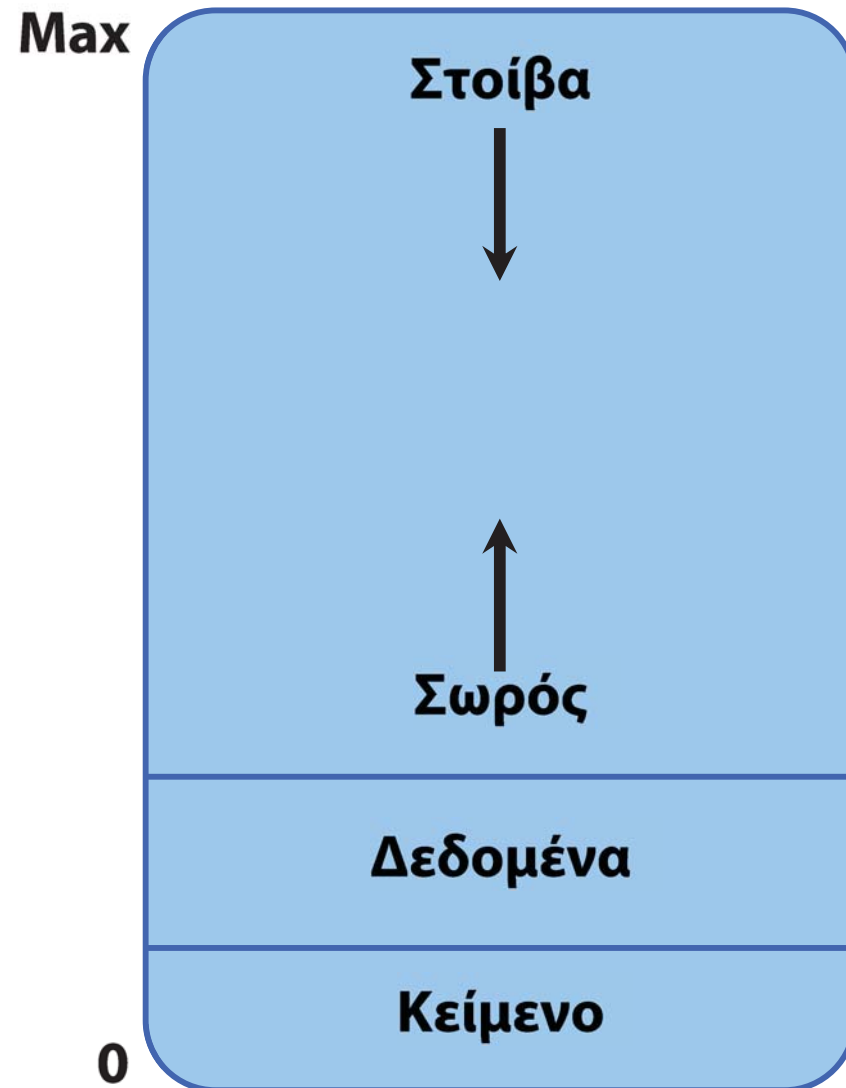


Δίσκος

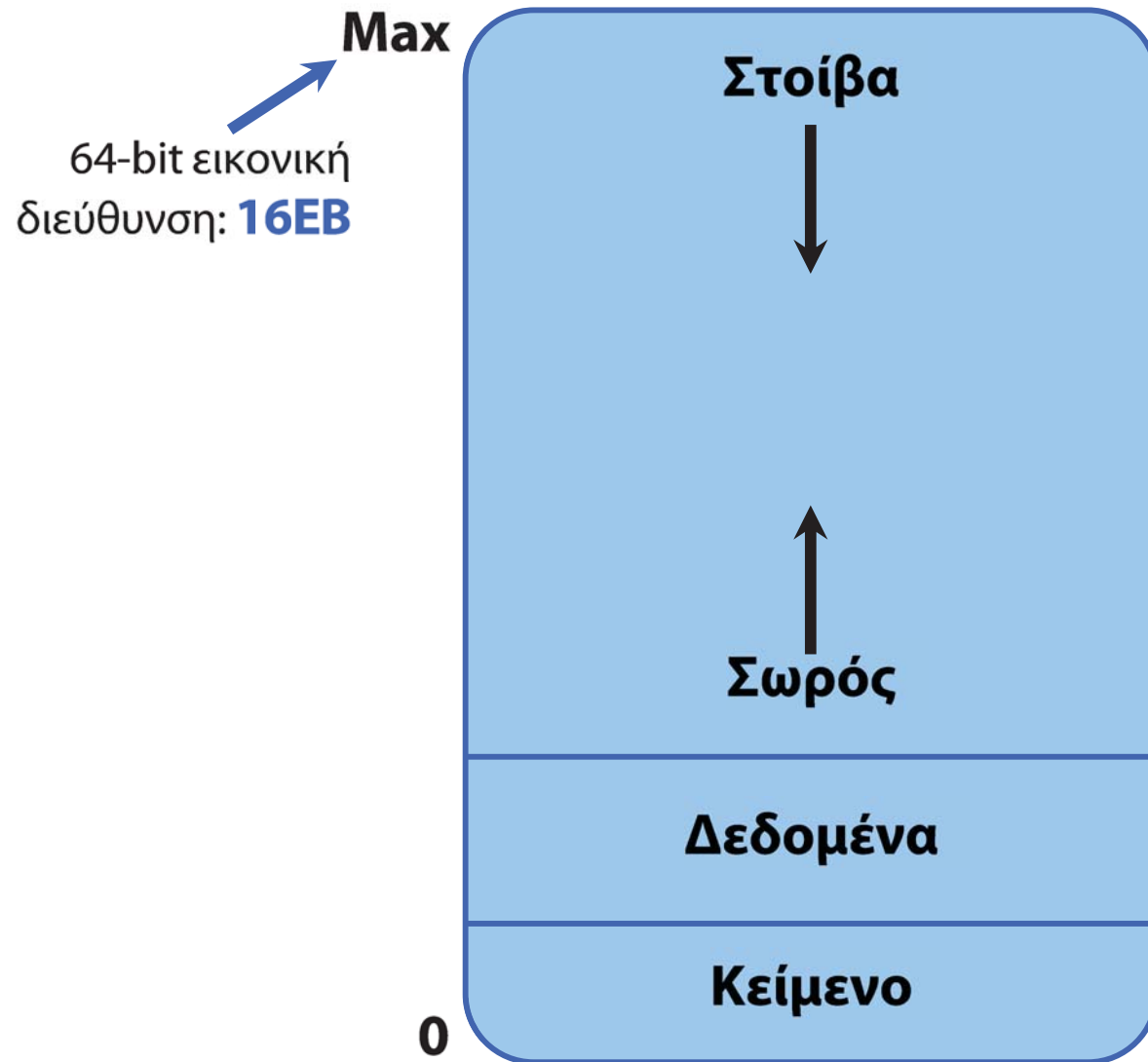
Χώρος Εικονικών Διευθύνσεων



Χώρος Εικονικών Διευθύνσεων



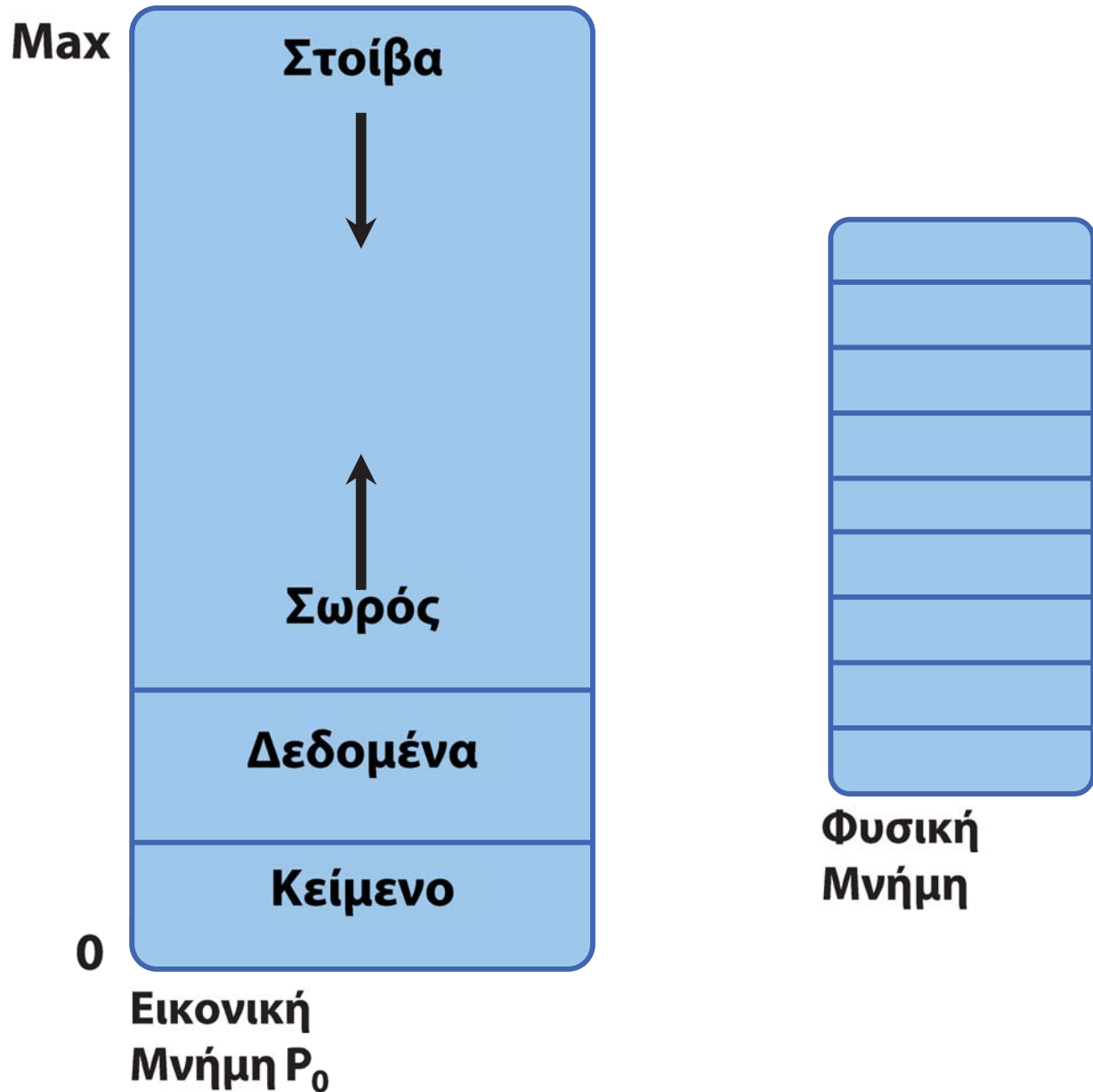
Χώρος Εικονικών Διευθύνσεων



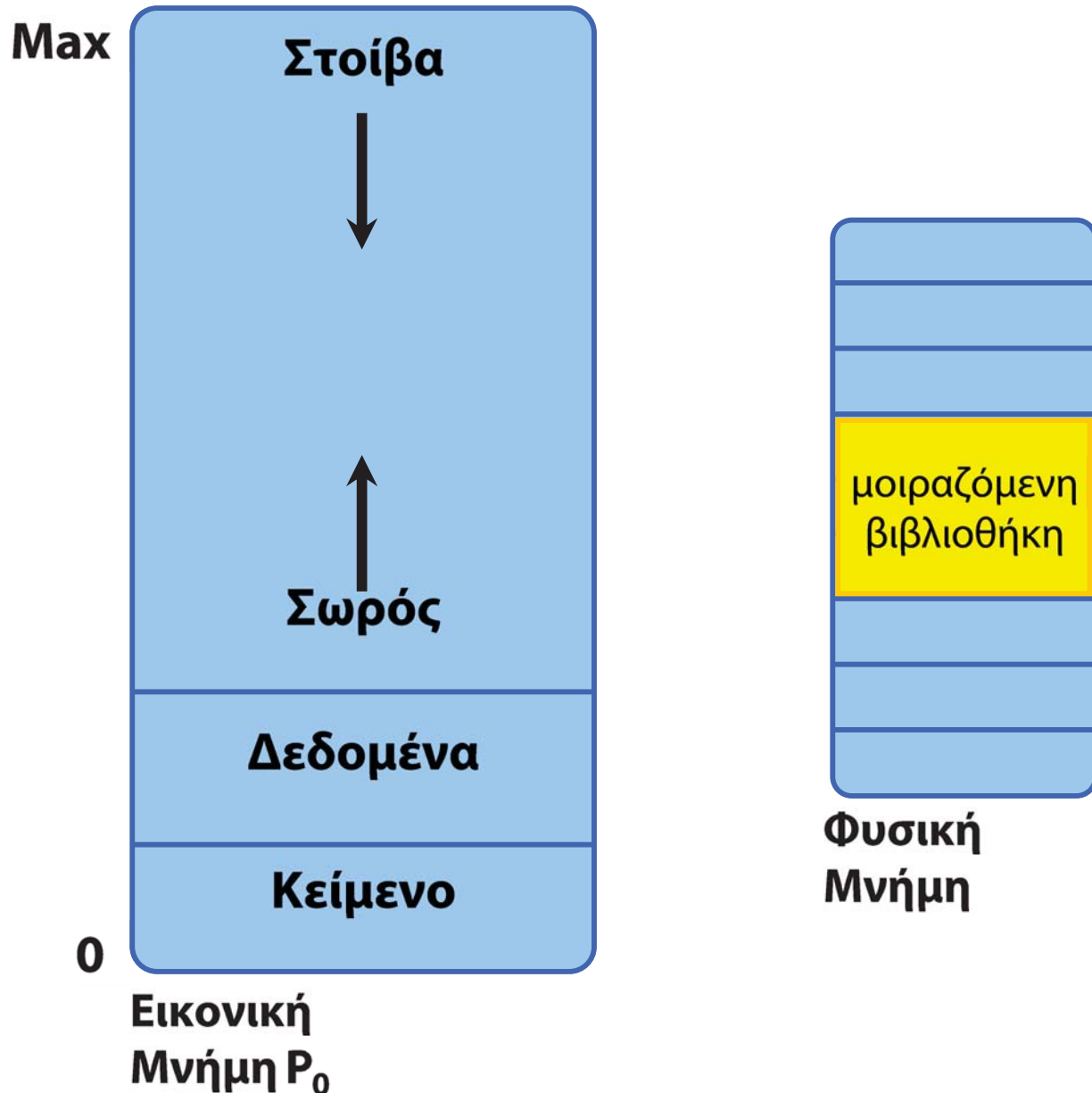
Εικονική Μνήμη - Σύνοψη

- ◆ 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

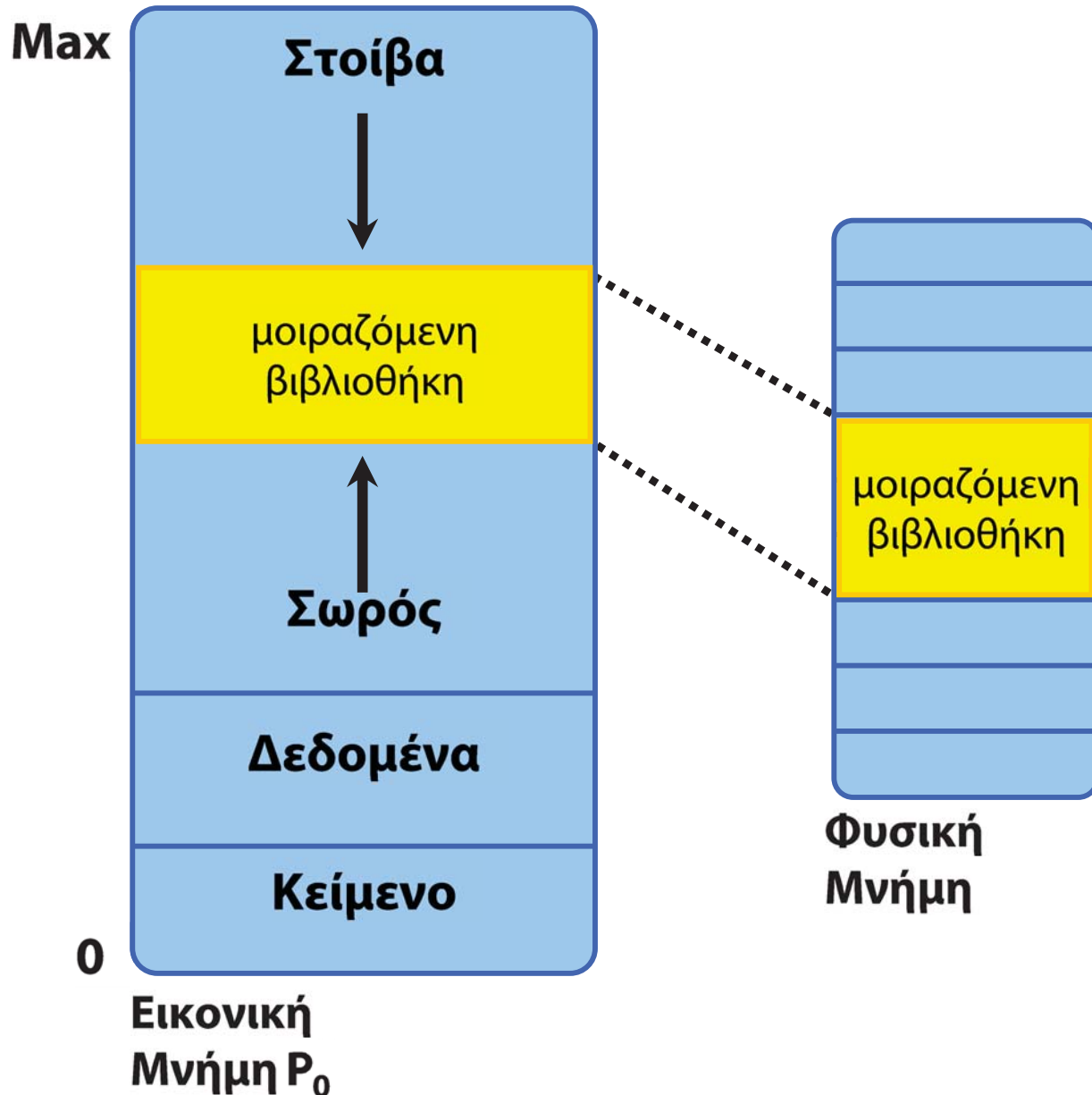
Δυναμική σύνδεση με μοιραζόμενες βιβλιοθήκες



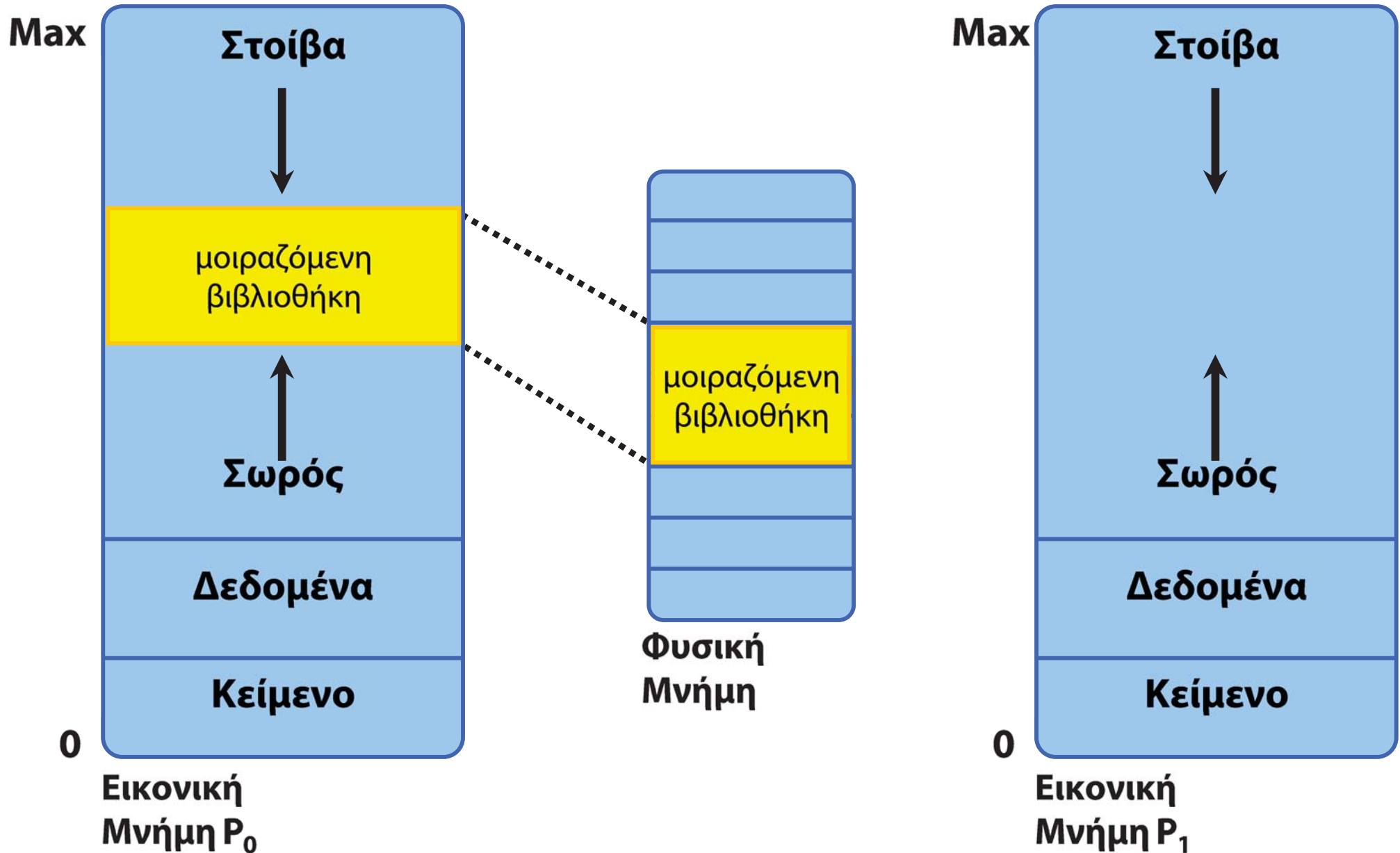
Δυναμική σύνδεση με μοιραζόμενες βιβλιοθήκες



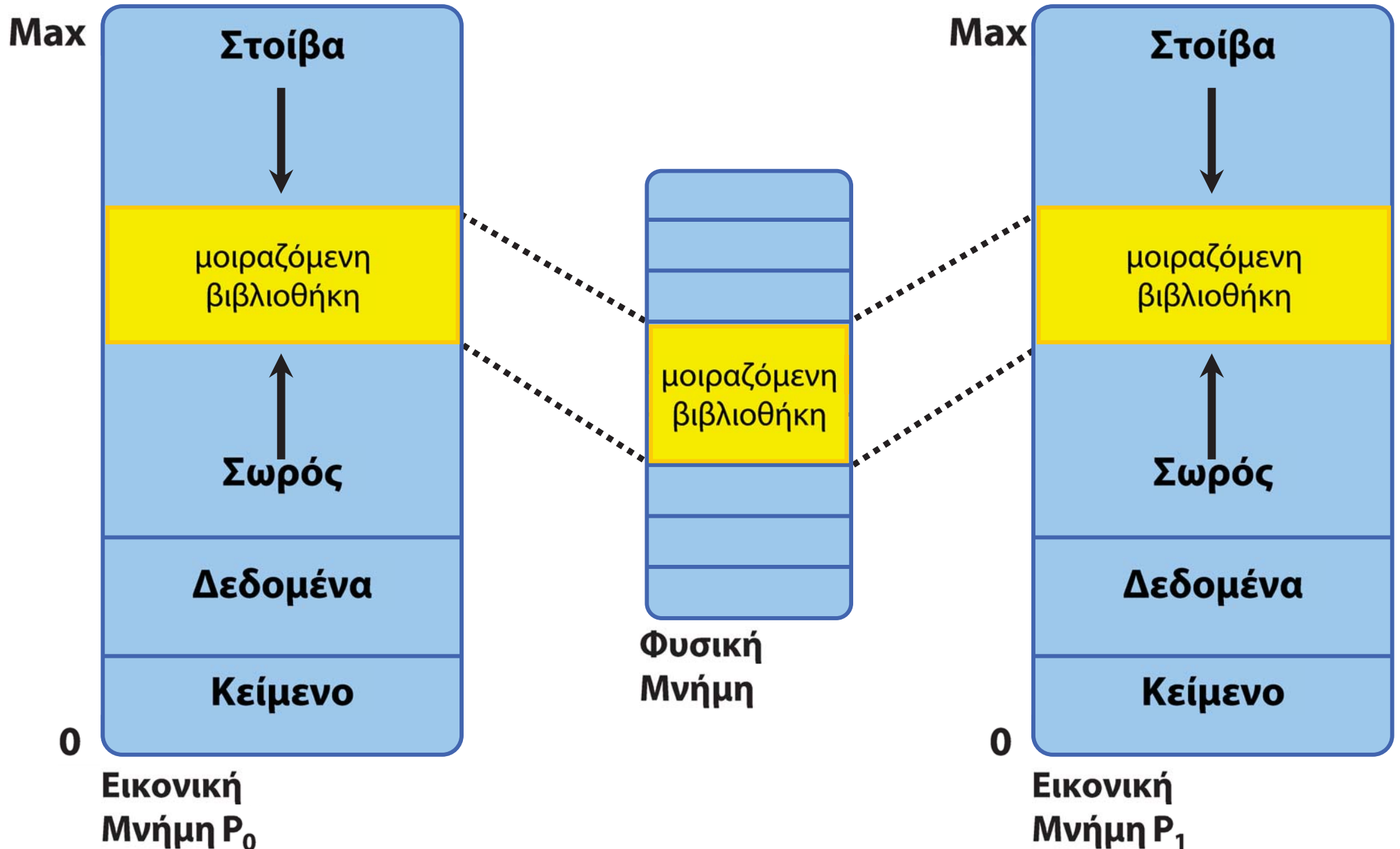
Δυναμική σύνδεση με μοιραζόμενες βιβλιοθήκες



Δυναμική σύνδεση με μοιραζόμενες βιβλιοθήκες



Δυναμική σύνδεση με μοιραζόμενες βιβλιοθήκες



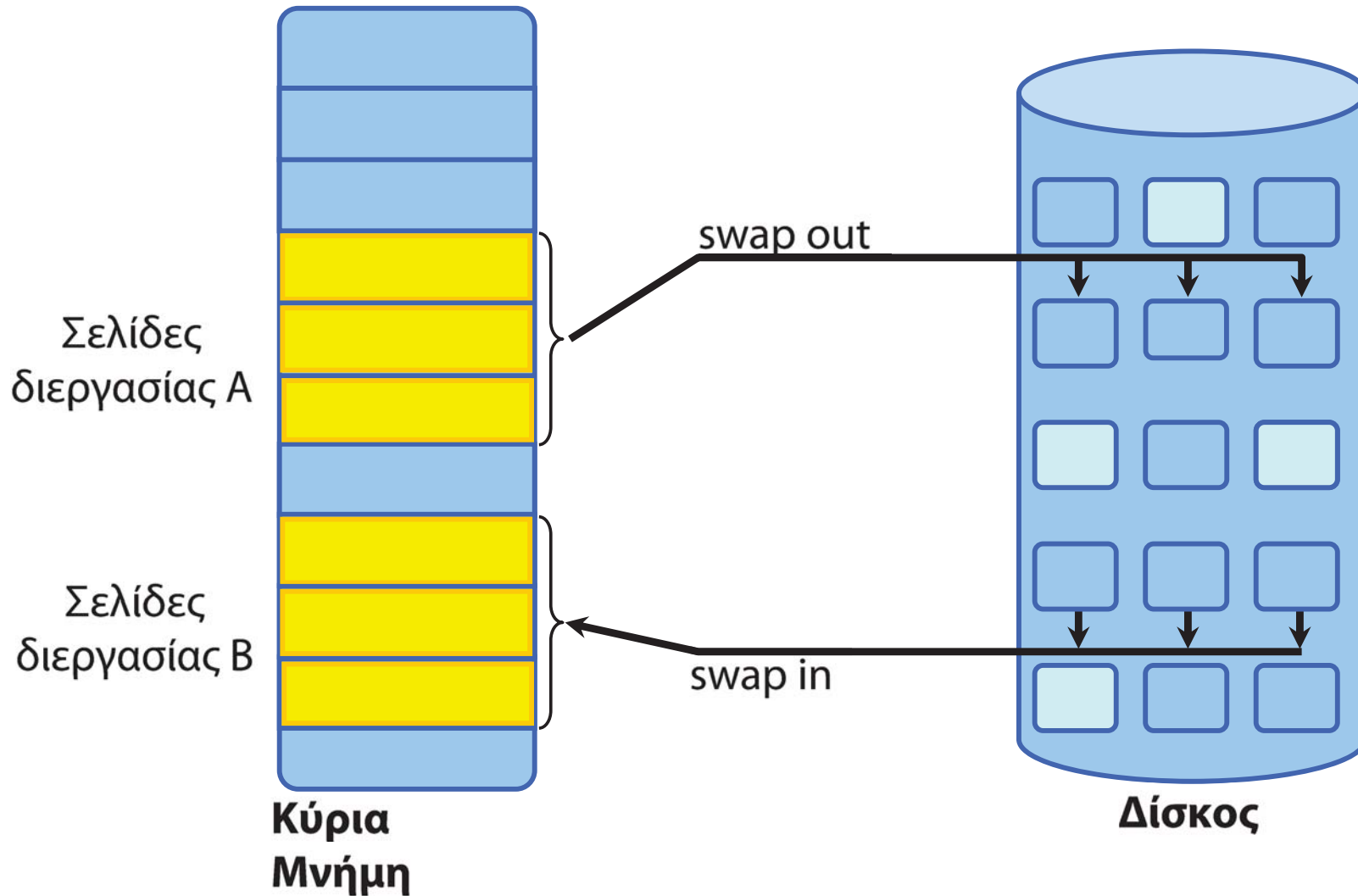
Εικονική Μνήμη - Σύνοψη

- ◆ 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

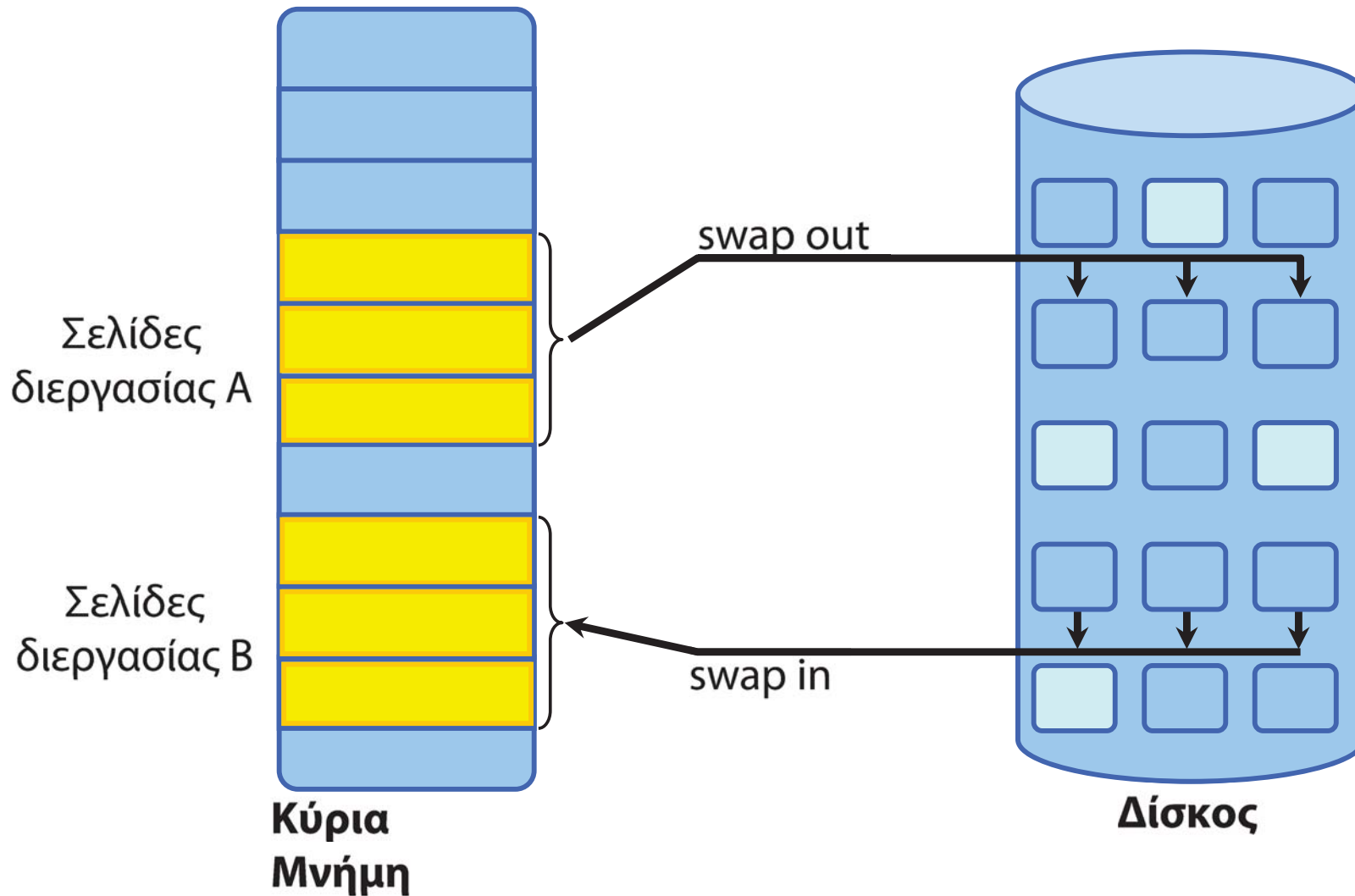
Σελιδοποίηση κατ' απαίτηση (Demand Paging)

- ◆ Σελίδες εικονικής μνήμης ζουν σε *φυσική μνήμη* και *δίσκο*
- ◆ Όταν μια διεργασία χρειαστεί συγκεκριμένη σελίδα (αναφορά), τότε φέρει 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 *pager*

Μεταφορά σελίδων από/προς Δίσκο

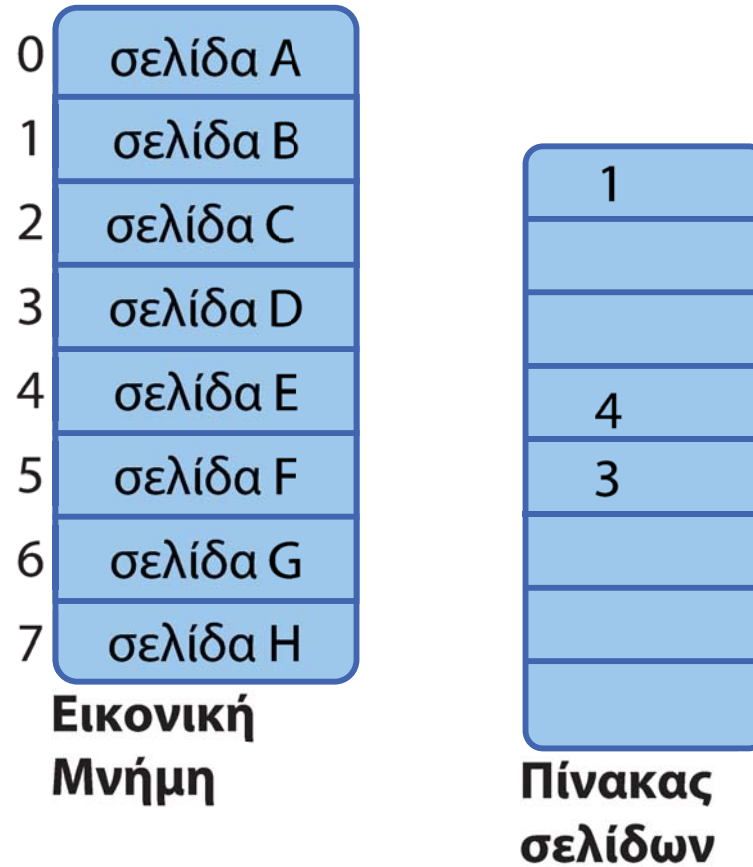


Μεταφορά σελίδων από/προς Δίσκο



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

Bit εγκυρότητας στον πίνακα σελίδων



Bit εγκυρότητας στον πίνακα σελίδων

0	σελίδα A	1	v
1	σελίδα B		i
2	σελίδα C		i
3	σελίδα D	4	v
4	σελίδα E	3	v
5	σελίδα F		i
6	σελίδα G		i
7	σελίδα H		i
Εικονική Μνήμη		Πίνακας σελίδων	

- ◆ Είναι κάθε αναφορά στο χώρο εικονικών διευθύνσεων *έγκυρη*;
- ◆ Επιπλέον bit στον πίνακα σελίδων: **V**alid – **I**nvalid
- ◆ Τι συμβαίνει αν γίνει αναφορά σε **I**nvalid σελίδα; **P**age **F**ault

Bit εγκυρότητας στον πίνακα σελίδων

0	σελίδα A	1	v
1	σελίδα B		i
2	σελίδα C		i
3	σελίδα D	4	v
4	σελίδα E	3	v
5	σελίδα F		i
6	σελίδα G		i
7	σελίδα H		i
Εικονική Μνήμη		Πίνακας σελίδων	

- ◆ Είναι κάθε αναφορά στο χώρο εικονικών διευθύνσεων *έγκυρη*;
- ◆ Επιπλέον bit στον πίνακα σελίδων: **V**alid – **I**nvalid
- ◆ Τι συμβαίνει αν γίνει αναφορά σε **I**nvalid σελίδα; **P**age **F**ault

Μερικές σελίδες είναι στο δίσκο

0	σελίδα A
1	σελίδα B
2	σελίδα C
3	σελίδα D
4	σελίδα E
5	σελίδα F
6	σελίδα G
7	σελίδα H

**Εικονική
Μνήμη**

Μερικές σελίδες είναι στο δίσκο

0	σελίδα A
1	σελίδα B
2	σελίδα C
3	σελίδα D
4	σελίδα E
5	σελίδα F
6	σελίδα G
7	σελίδα H

**Εικονική
Μνήμη**

1	v
	i
	i
4	v
3	v
	i
	i
	i

**Πίνακας
σελίδων**

Μερικές σελίδες είναι στο δίσκο

0	σελίδα A
1	σελίδα B
2	σελίδα C
3	σελίδα D
4	σελίδα E
5	σελίδα F
6	σελίδα G
7	σελίδα H

**Εικονική
Μνήμη**

1	v
	i
	i
4	v
3	v
	i
	i
	i

**Πίνακας
σελίδων**

0	
1	A
2	
3	E
4	D
5	
6	
7	

**Φυσική
Μνήμη**

Μερικές σελίδες είναι στο δίσκο

0	σελίδα A
1	σελίδα B
2	σελίδα C
3	σελίδα D
4	σελίδα E
5	σελίδα F
6	σελίδα G
7	σελίδα H

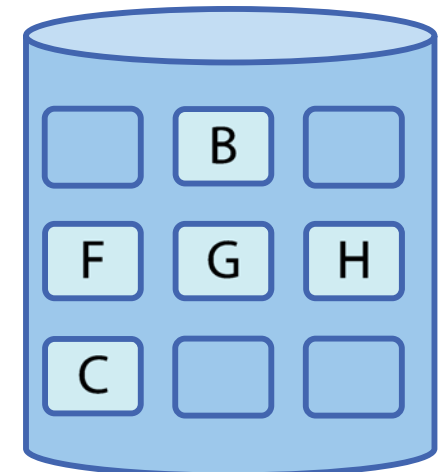
**Εικονική
Μνήμη**

1	v
	i
	i
4	v
3	v
	i
	i
	i

**Πίνακας
σελίδων**

0	
1	A
2	
3	E
4	D
5	
6	
7	

**Φυσική
Μνήμη**



Δίσκος

Μερικές σελίδες είναι στο δίσκο

0	σελίδα A
1	σελίδα B
2	σελίδα C
3	σελίδα D
4	σελίδα E
5	σελίδα F
6	σελίδα G
7	σελίδα H

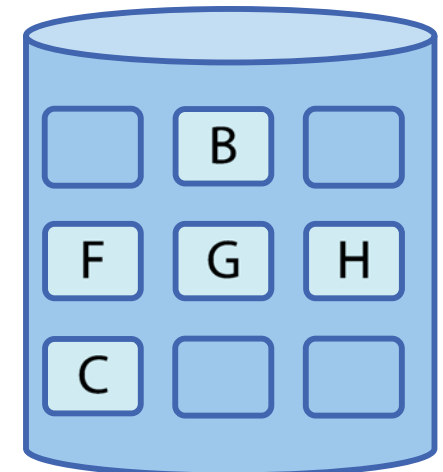
**Εικονική
Μνήμη**

1	v
	i
	i
4	v
3	v
	i
	i
	i

**Πίνακας
σελίδων**

0	
1	A
2	
3	E
4	D
5	
6	
7	

**Φυσική
Μνήμη**



Δίσκος

- ◆ Ποιος ξέρει πού είναι οι σελίδες στο δίσκο;
➔ Το ΛΣ
- ◆ Αναφορά σε σελίδα που δεν υπάρχει στη μνήμη;
➔ Page fault!

Μερικές σελίδες είναι στο δίσκο

0	σελίδα A
1	σελίδα B
2	σελίδα C
3	σελίδα D
4	σελίδα E
5	σελίδα F
6	σελίδα G
7	σελίδα H

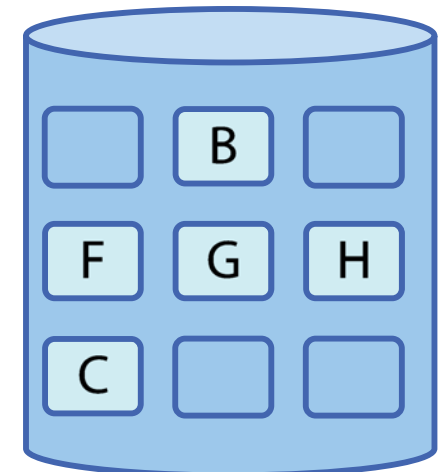
**Εικονική
Μνήμη**

1	v
	i
	i
4	v
3	v
	i
	i
	i

**Πίνακας
σελίδων**

0	
1	A
2	
3	E
4	D
5	
6	
7	

**Φυσική
Μνήμη**



Δίσκος

- ◆ Ποιος ξέρει πού είναι οι σελίδες στο δίσκο;
➔ Το ΛΣ
- ◆ Αναφορά σε σελίδα που δεν υπάρχει στη μνήμη;
➔ Page fault!

Μερικές σελίδες είναι στο δίσκο

0	σελίδα A
1	σελίδα B
2	σελίδα C
3	σελίδα D
4	σελίδα E
5	σελίδα F
6	σελίδα G
7	σελίδα H

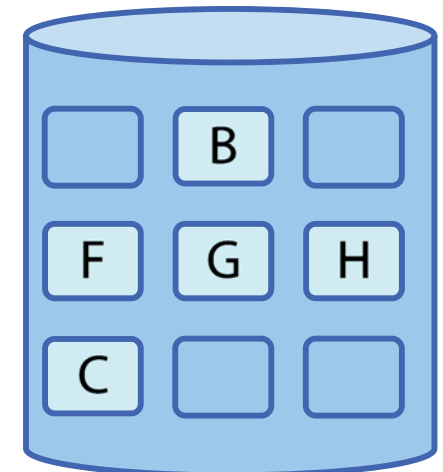
**Εικονική
Μνήμη**

1	v
	i
	i
4	v
3	v
	i
	i
	i

**Πίνακας
σελίδων**

0	
1	A
2	
3	E
4	D
5	
6	
7	

**Φυσική
Μνήμη**



Δίσκος

- ◆ Ποιος ξέρει πού είναι οι σελίδες στο δίσκο;
➔ Το ΛΣ
- ◆ Αναφορά σε σελίδα που δεν υπάρχει στη μνήμη;
➔ Page fault!

Μερικές σελίδες είναι στο δίσκο

0	σελίδα A
1	σελίδα B
2	σελίδα C
3	σελίδα D
4	σελίδα E
5	σελίδα F
6	σελίδα G
7	σελίδα H

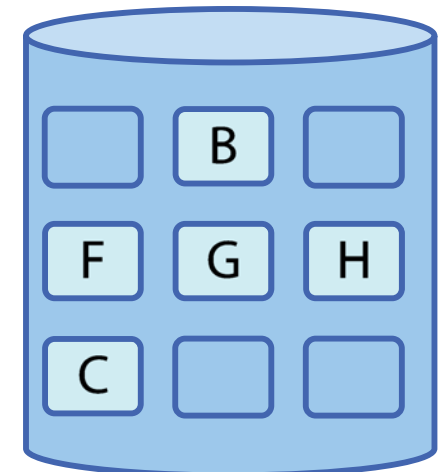
**Εικονική
Μνήμη**

1	v
	i
	i
4	v
3	v
	i
	i
	i

**Πίνακας
σελίδων**

0	
1	A
2	
3	E
4	D
5	
6	
7	

**Φυσική
Μνήμη**



Δίσκος

- ◆ Ποιος ξέρει πού είναι οι σελίδες στο δίσκο;
➔ Το ΛΣ
- ◆ Αναφορά σε σελίδα που δεν υπάρχει στη μνήμη;
➔ Page fault!

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

Σφάλμα Σελίδας – Page Fault (1)



Σφάλμα Σελίδας – Page Fault (1)

- ◆ **Εξαίρεση:** Το ΛΣ ξυπνάει
- ◆ Το πρόγραμμα έκανε αναφορά σε άκυρη σελίδα
- ◆ **Γιατί;** Το ΛΣ μελετά τον πίνακα 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;

Σφάλμα Σελίδας – Page Fault (1)

- ◆ **Εξαίρεση:** Το ΛΣ ξυπνάει
- ◆ Το πρόγραμμα έκανε αναφορά σε άκυρη σελίδα
- ◆ **Γιατί;** Το ΛΣ μελετά τον πίνακα 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;

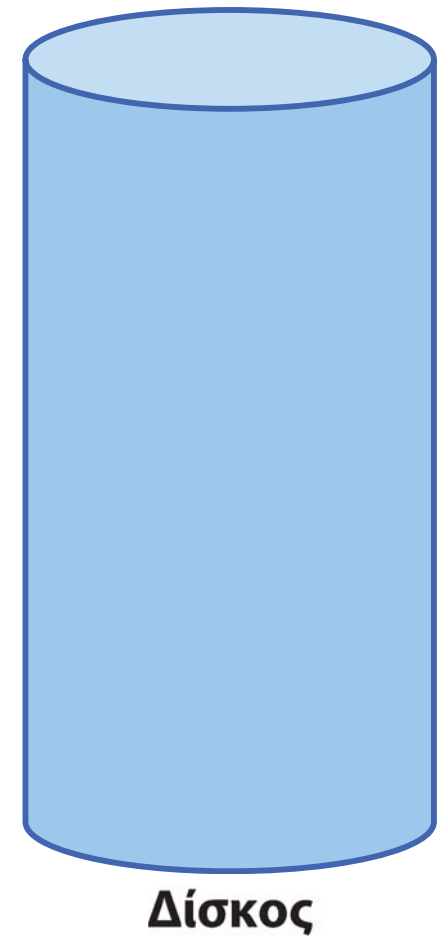
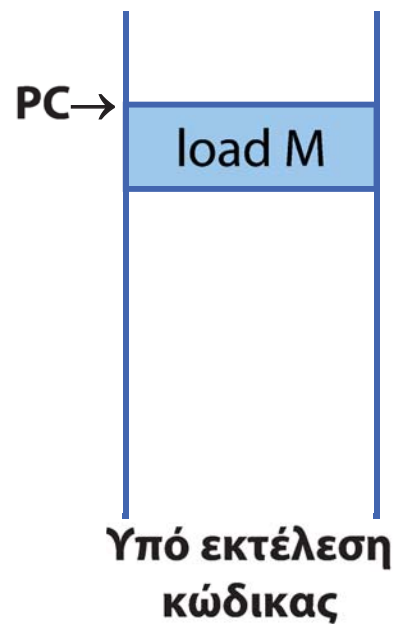
Σφάλμα Σελίδας – Page Fault (1)

- ◆ **Εξαίρεση:** Το ΛΣ ξυπνάει
- ◆ Το πρόγραμμα έκανε αναφορά σε άκυρη σελίδα
- ◆ **Γιατί;** Το ΛΣ μελετά τον πίνακα 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;

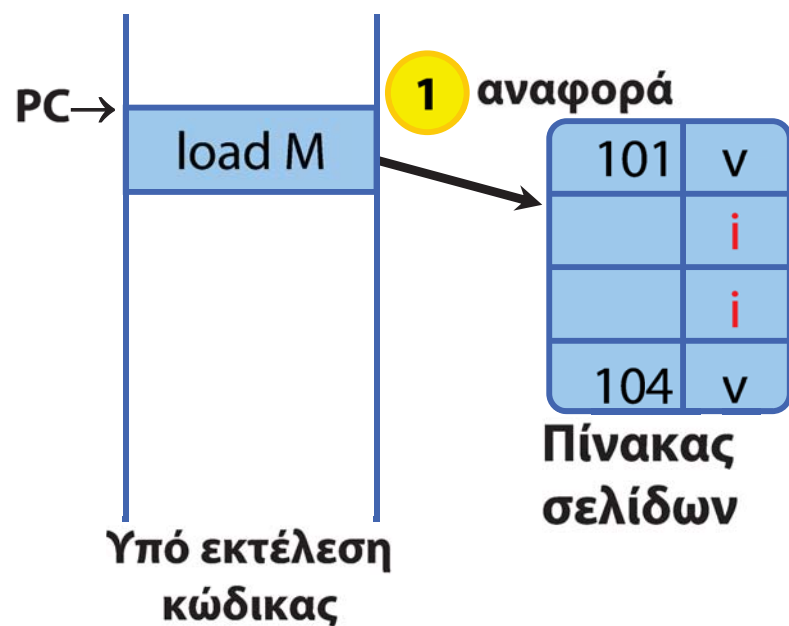
Εξυπηρέτηση σφάλματος σελίδας (1)

- ◆ Βρες ένα ελεύθερο πλαίσιο μνήμης
 - ➔ Υπάρχει πάντα;
- ◆ Φέρε τα δεδομένα της σελίδας από το δίσκο σε αυτό το πλαίσιο
- ◆ 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 Κύρια Μνήμη;

Εξυπηρέτηση σφάλματος σελίδας (2)



Εξυπηρέτηση σφάλματος σελίδας (2)

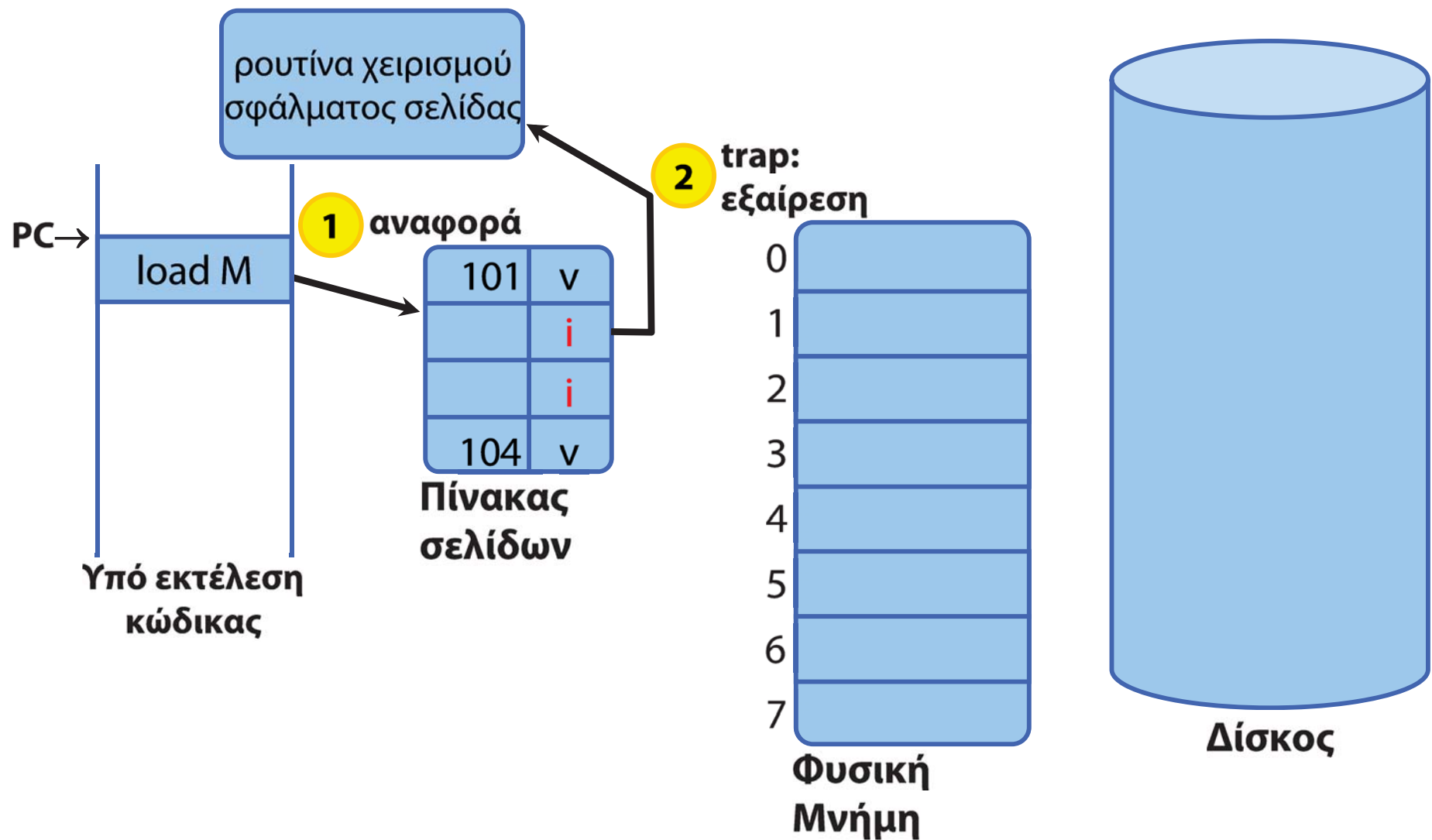


Φυσική
Μνήμη

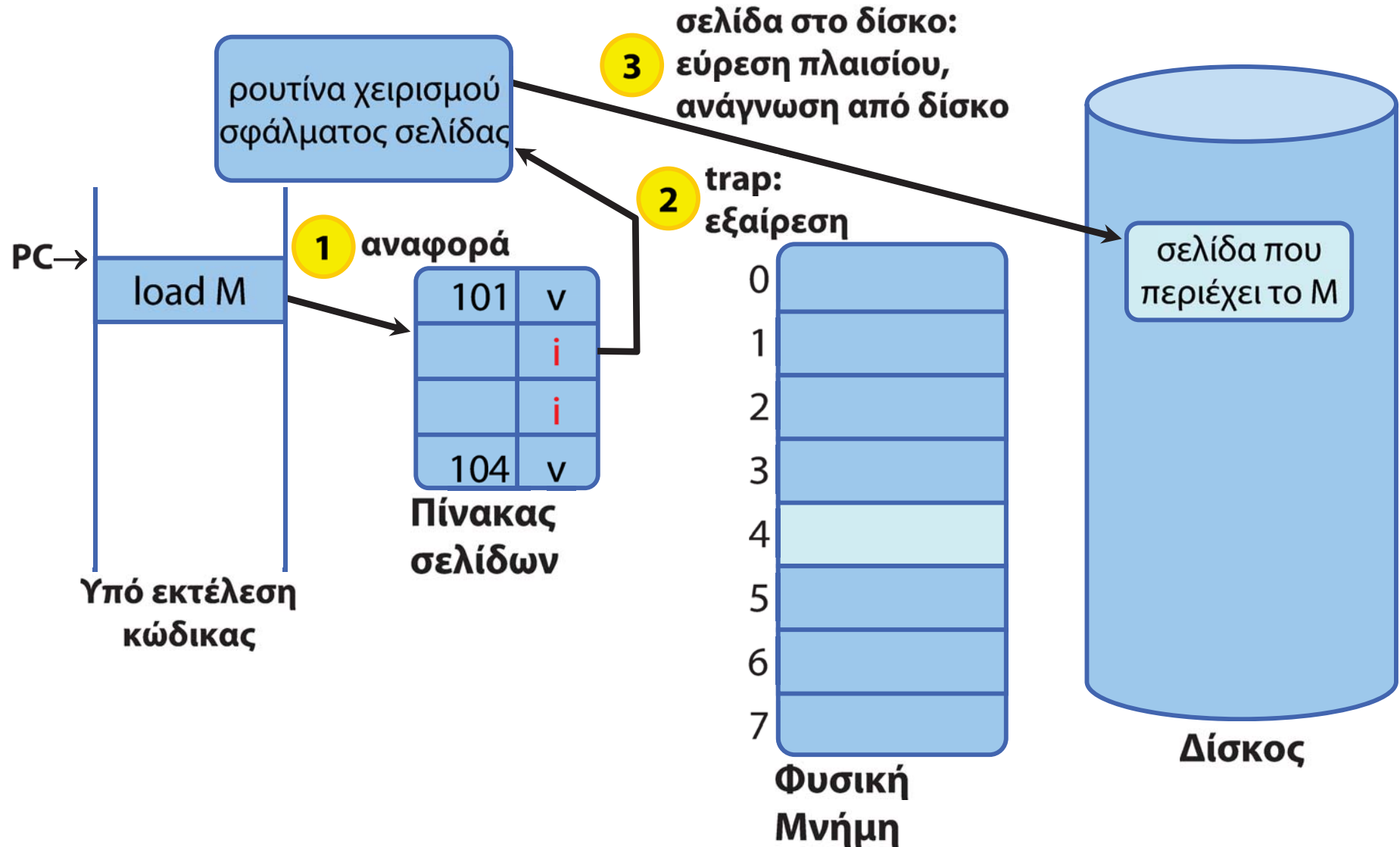


Δίσκος

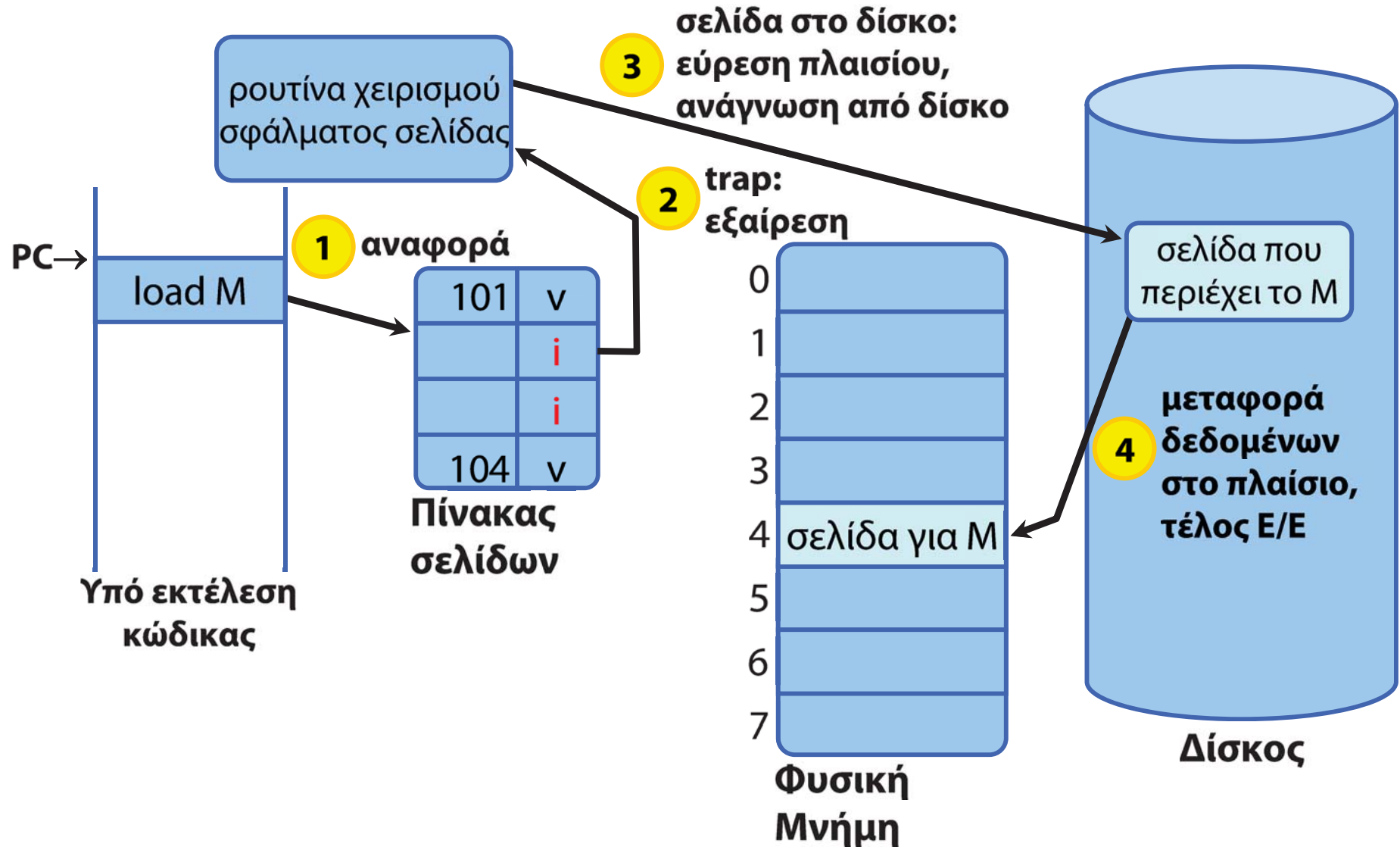
Εξυπηρέτηση σφάλματος σελίδας (2)



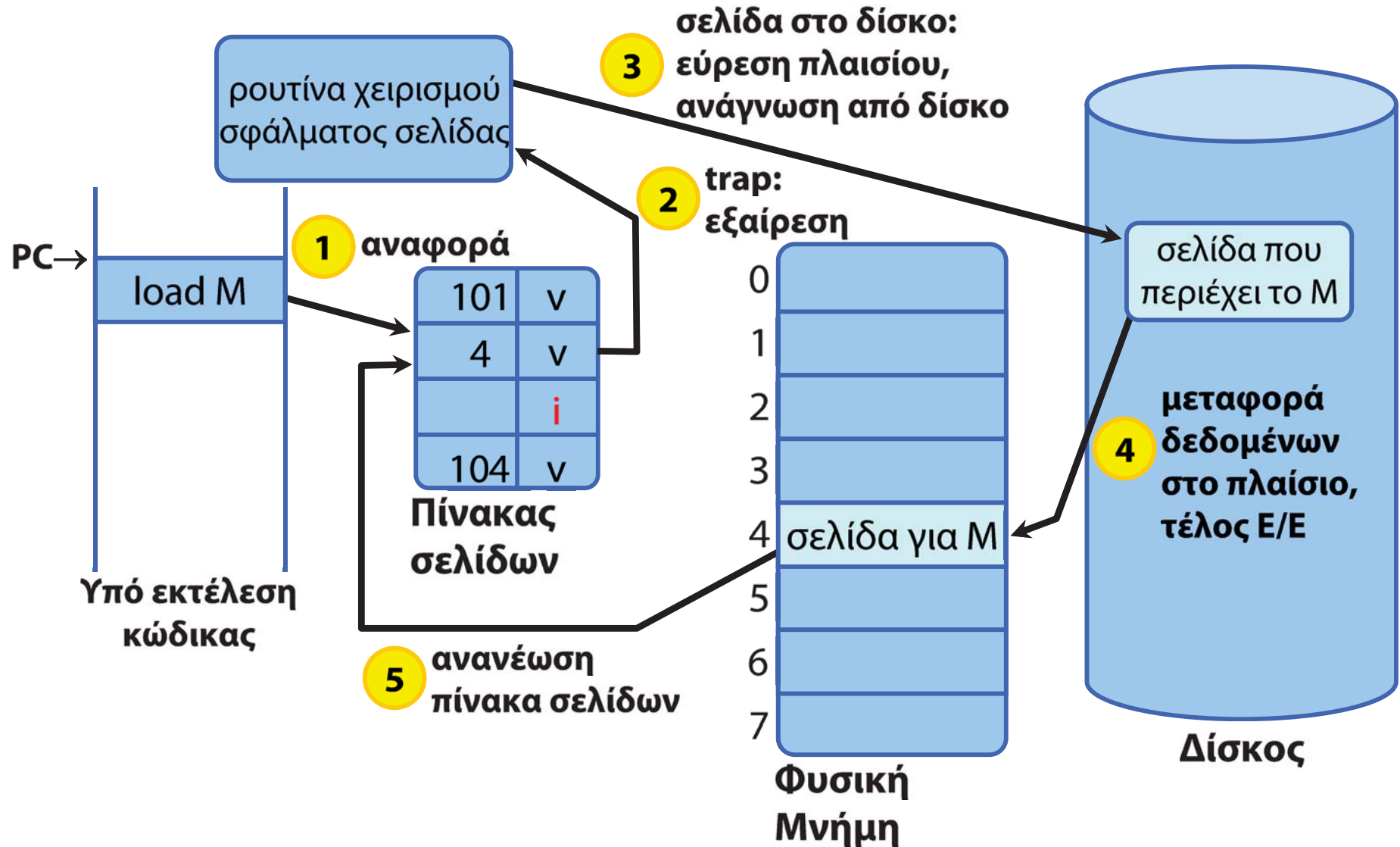
Εξυπηρέτηση σφάλματος σελίδας (2)



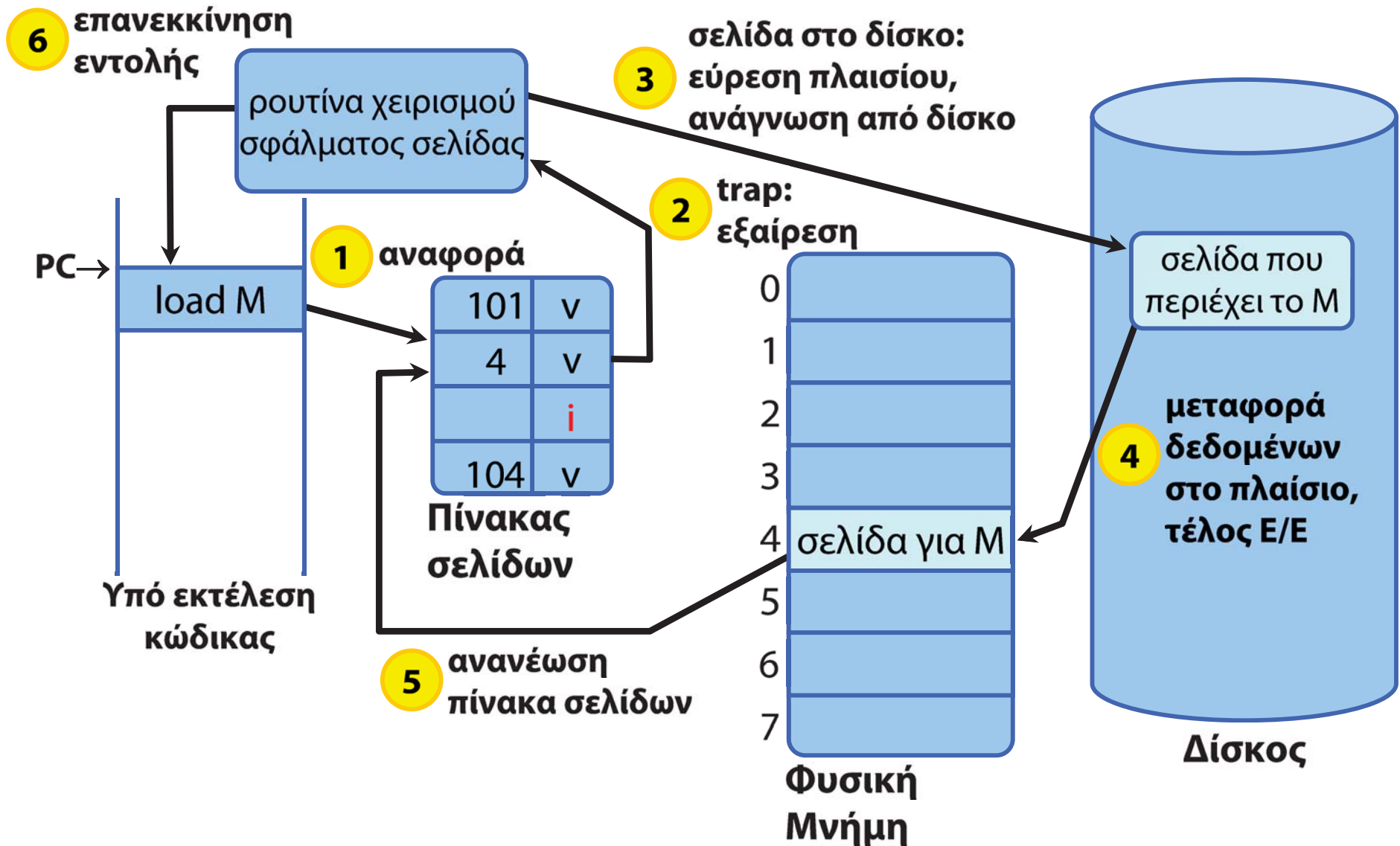
Εξυπηρέτηση σφάλματος σελίδας (2)



Εξυπηρέτηση σφάλματος σελίδας (2)



Εξυπηρέτηση σφάλματος σελίδας (2)



Εξυπηρέτηση σφάλματος σελίδας (3)



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 E/E
 - ➔ 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 E/E
 - ➔ όταν ο δίσκος ολοκληρώσει τη μεταφορά → διακοπή
 - ➔ το ΛΣ διορθώνει τον πίνακα σελίδων
 - ➔ η διεργασία γίνεται **έτοιμη**

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 E/E
 - ➔ 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 E/E
 - ➔ όταν ο δίσκος ολοκληρώσει τη μεταφορά → διακοπή
 - ➔ το ΛΣ διορθώνει τον πίνακα σελίδων
 - ➔ η διεργασία γίνεται **έτοιμη**

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 E/E
 - ➔ 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 E/E
 - ➔ όταν ο δίσκος ολοκληρώσει τη μεταφορά → διακοπή
 - ➔ το ΛΣ διορθώνει τον πίνακα σελίδων
 - ➔ η διεργασία γίνεται **έτοιμη**

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 E/E
 - ➔ 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 E/E
 - ➔ 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
 - $EAT = (1 - p) \times t_m +$
 $p \times (\text{κόστος σφάλματος σελίδας} +$
 $\text{κόστος swap out} + \text{κόστος swap in} +$
 $\text{κόστος επανεκκίνησης})$
 - ➔ 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
 - ➔ 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

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



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;
- ◆ *Όχι!* Σε συστήματα εικονικής μνήμης
 - ➔ Αντιγραφή κατά την Εγγραφή – Copy-on-Write (COW)

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;
- ◆ *Όχι!* Σε συστήματα εικονικής μνήμης
 - ➔ Αντιγραφή κατά την Εγγραφή – Copy-on-Write (COW)

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;
- ◆ *Όχι!* Σε συστήματα εικονικής μνήμης
 - ➔ Αντιγραφή κατά την Εγγραφή – Copy-on-Write (COW)

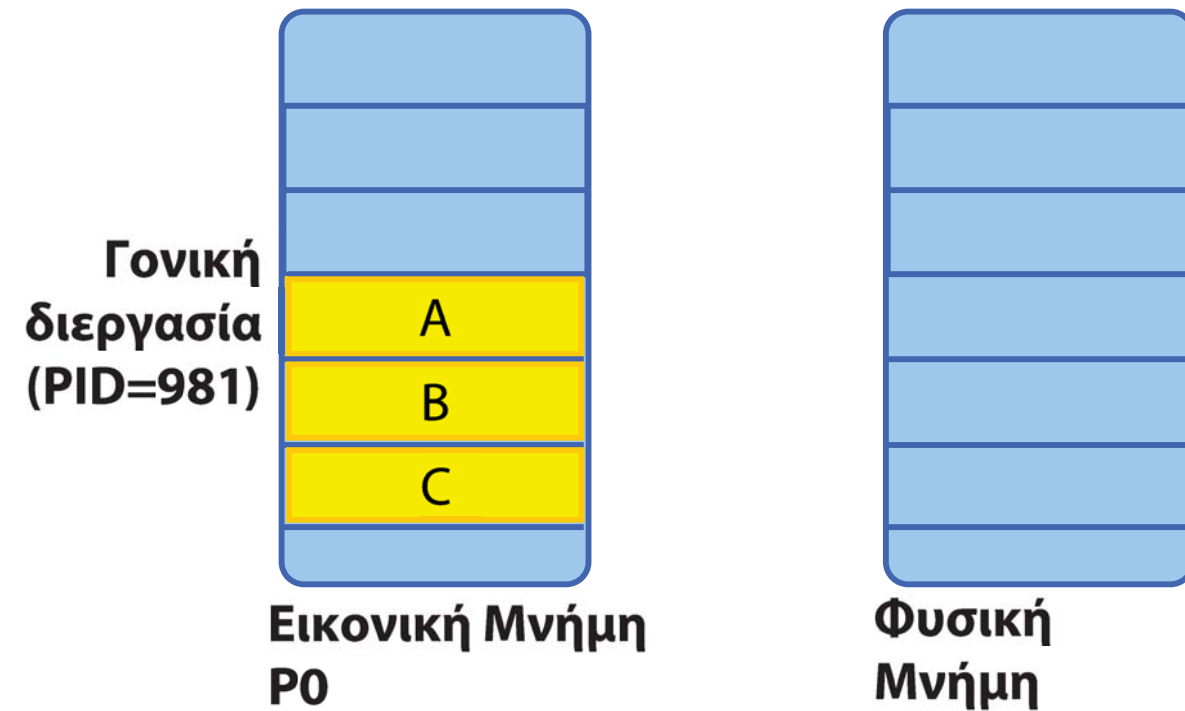


Δημιουργία νέας διεργασίας – π.χ. **fork()**

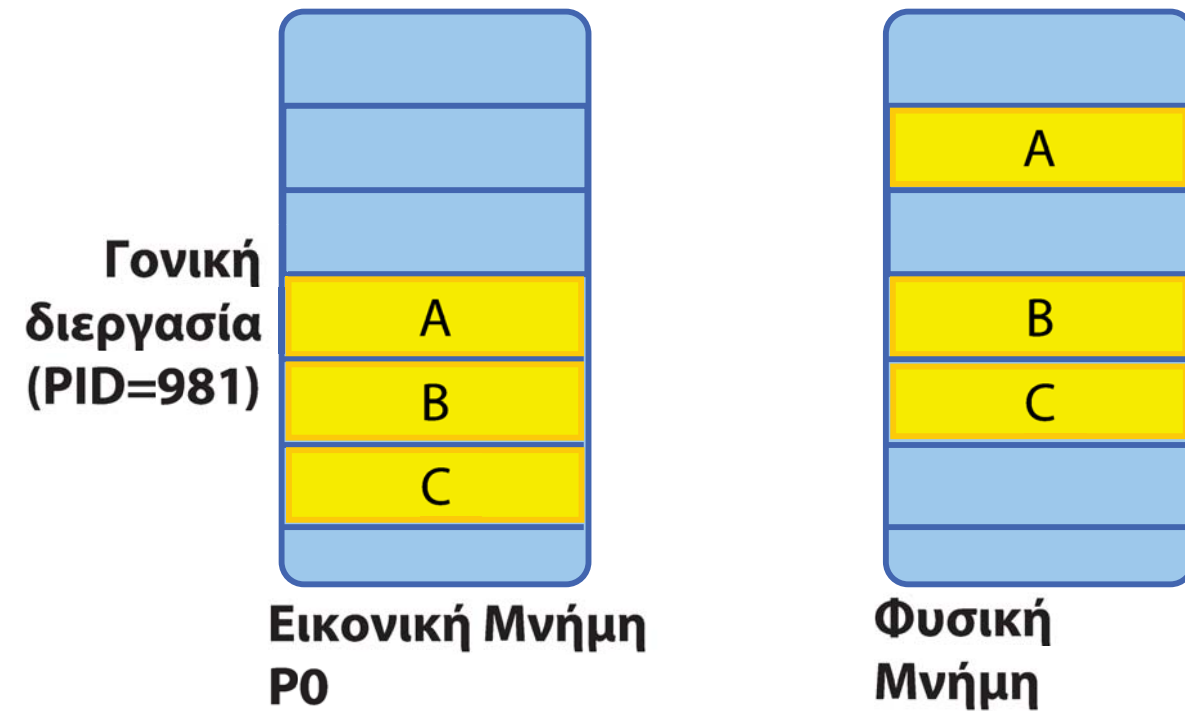
- ◆ Η νέα διεργασία είναι *αντίγραφο* της παλιάς
- ◆ Κληρονομεί όλα τα ανοιχτά αρχεία και αντίγραφο της μνήμης της αρχικής διεργασίας
- ◆ Από εκεί και πέρα, εκτελείται ανεξάρτητα
- ◆ Πόσο κοστίζει η δημιουργία της;
 - ➔ Αντιγράφεται όλη η μνήμη της παλιάς διεργασίας
 - ➔ Πάντα;
- ◆ *Όχι!* Σε συστήματα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



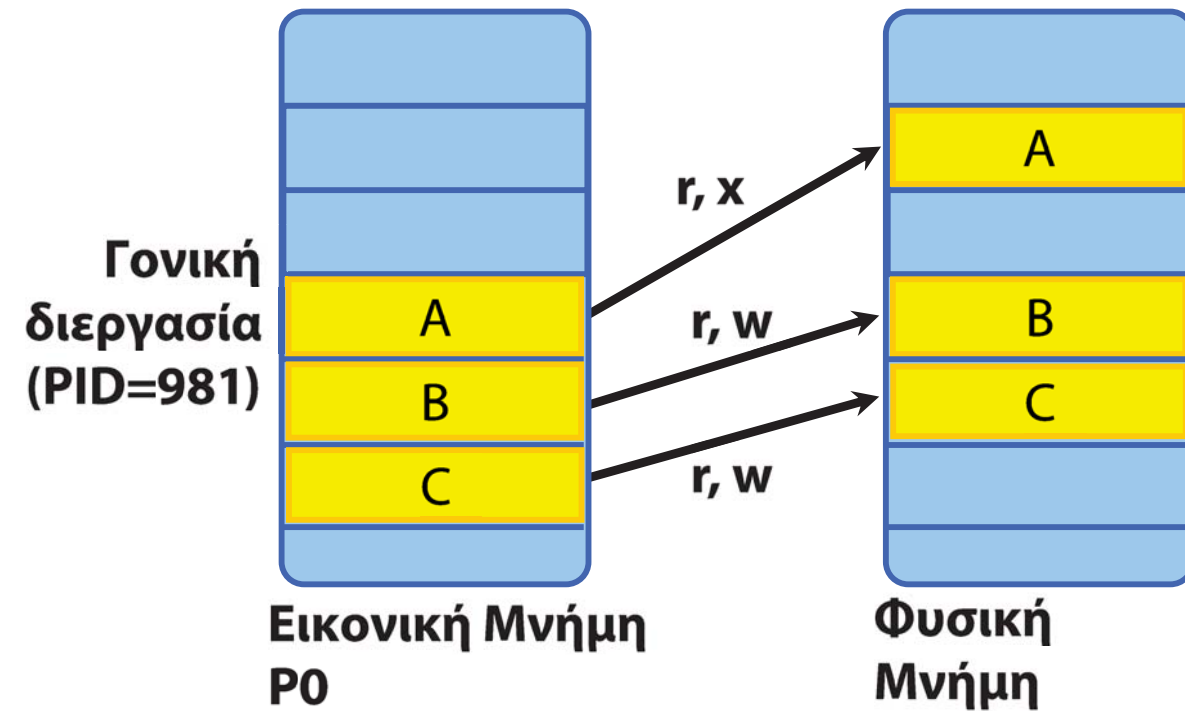
Copy-on-Write



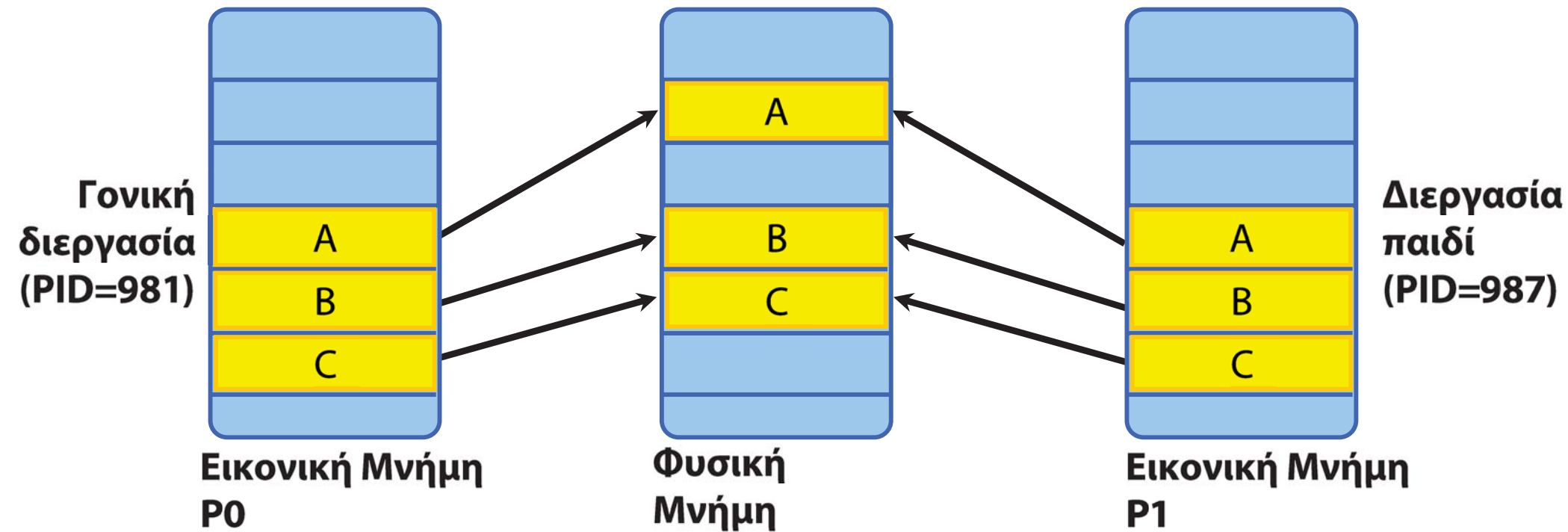
Copy-on-Write



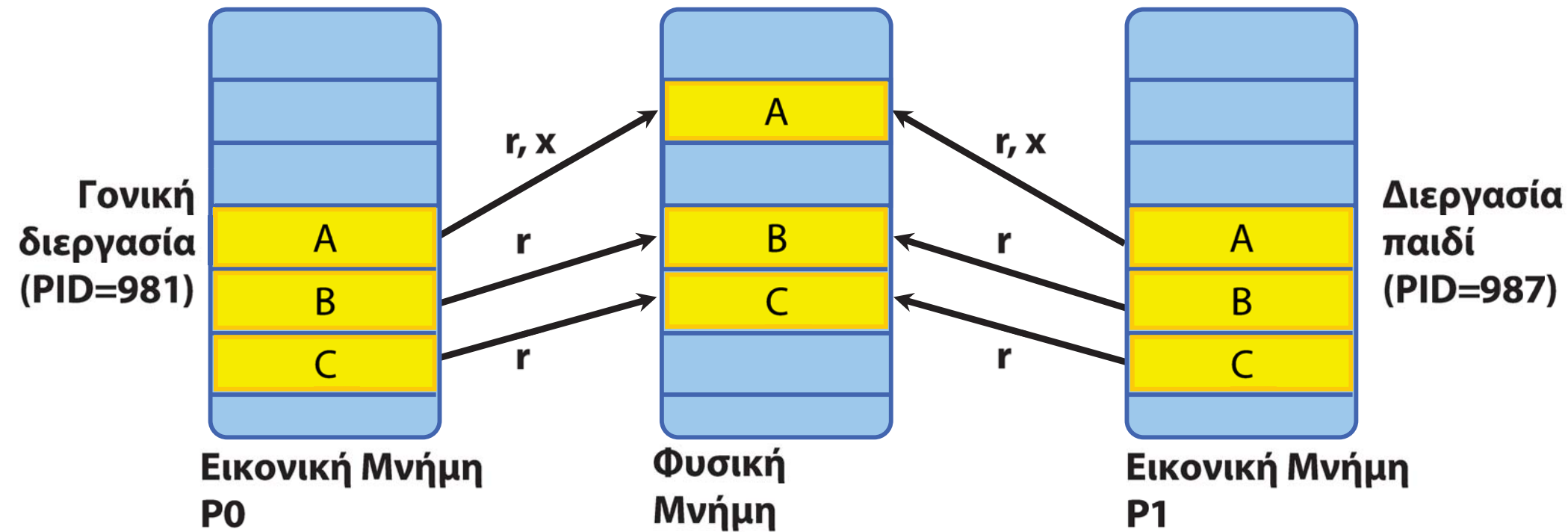
Copy-on-Write



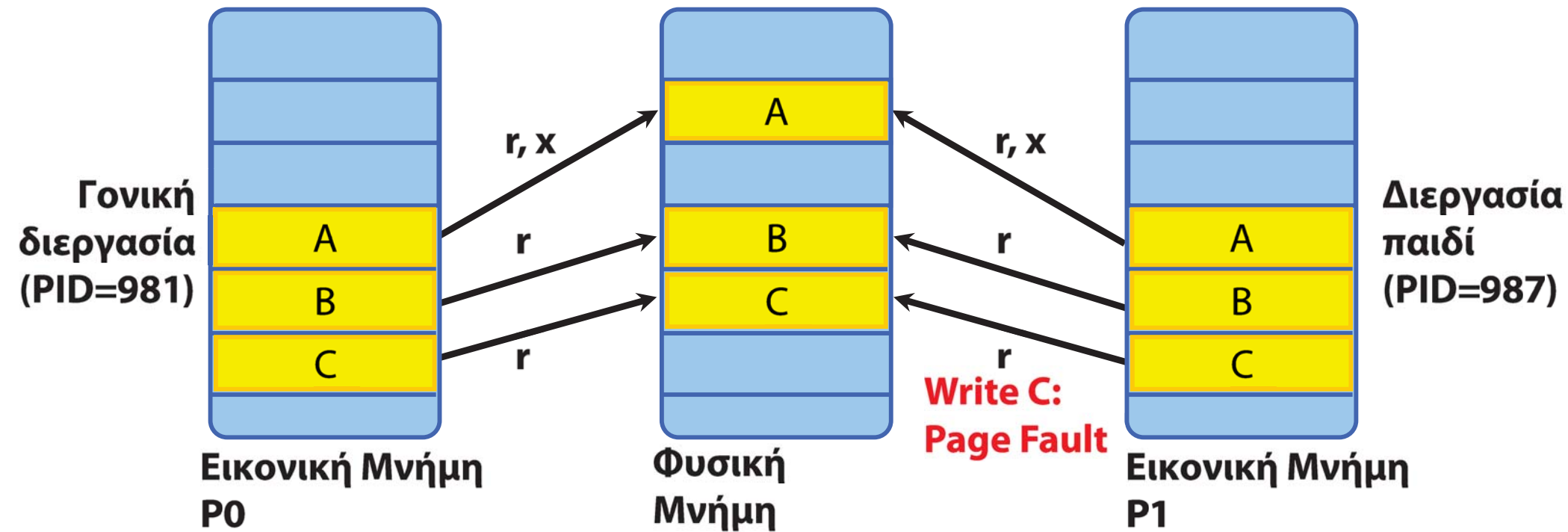
Copy-on-Write



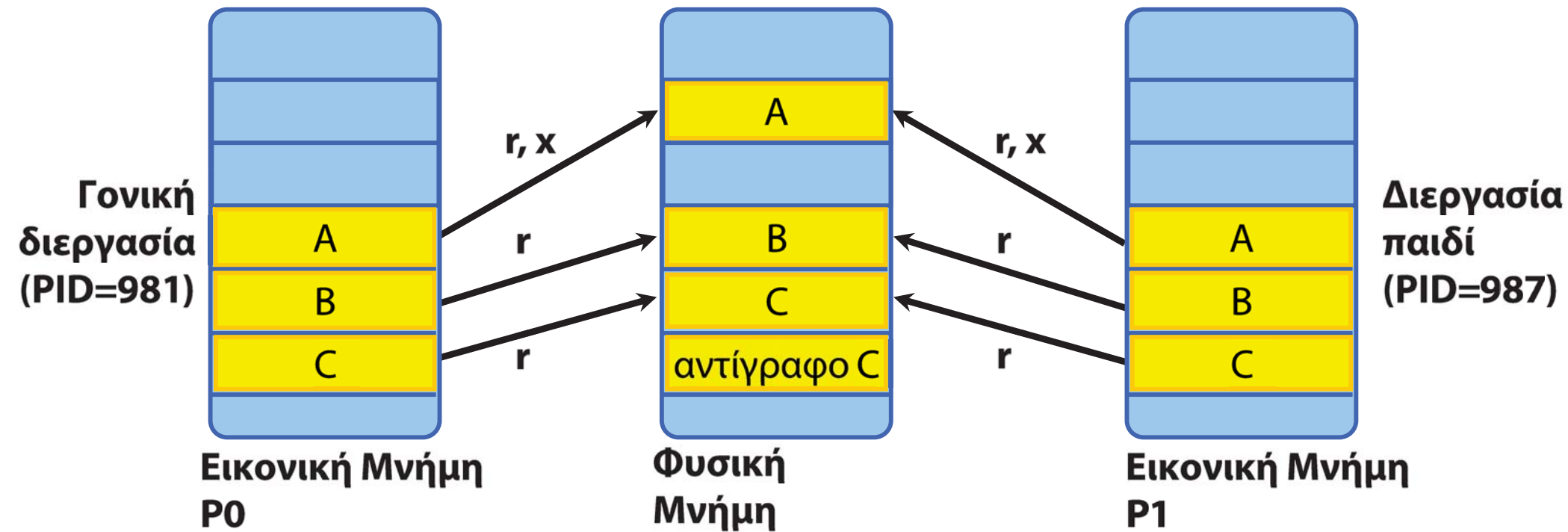
Copy-on-Write



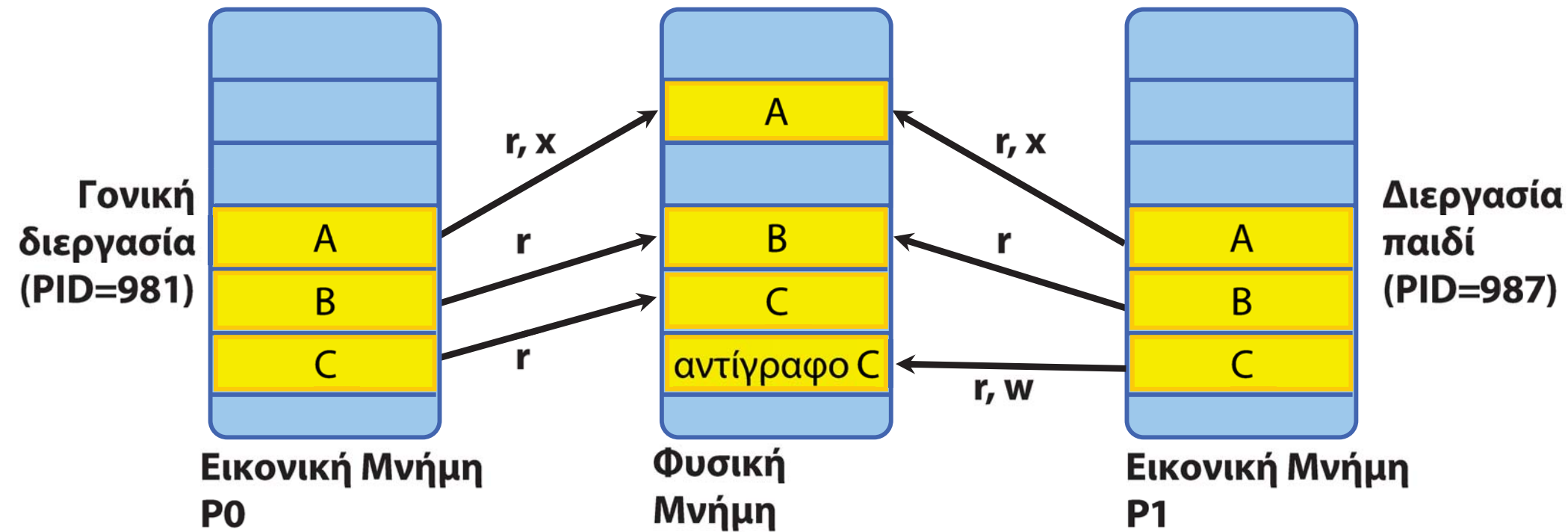
Copy-on-Write



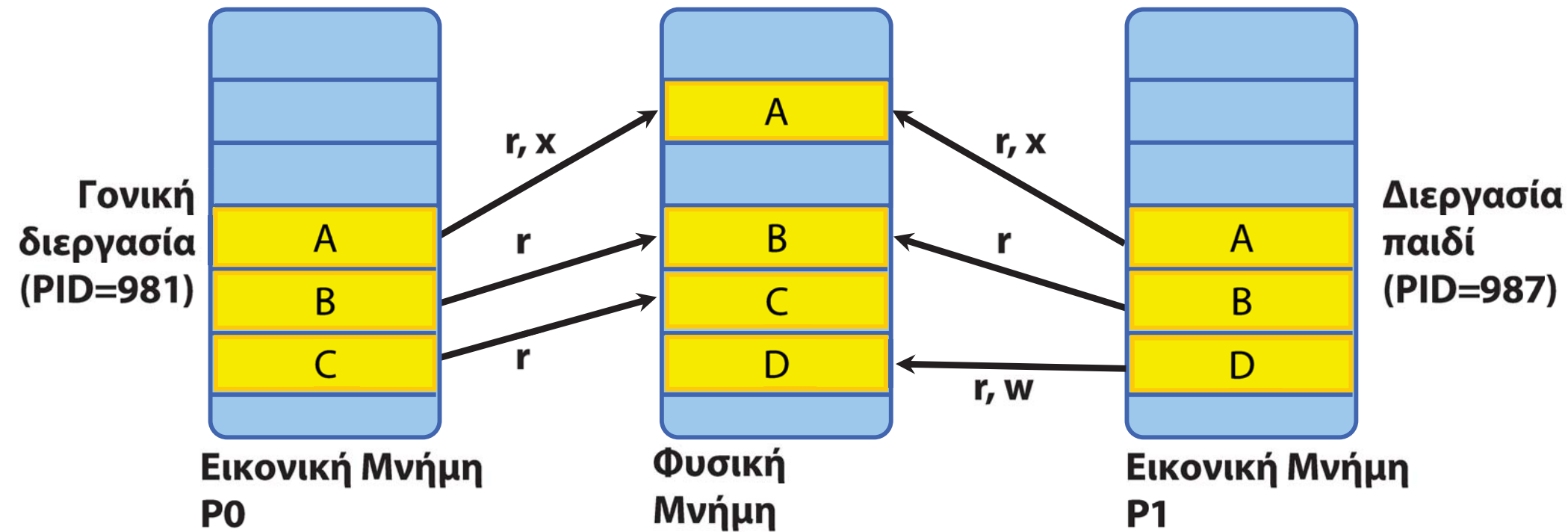
Copy-on-Write



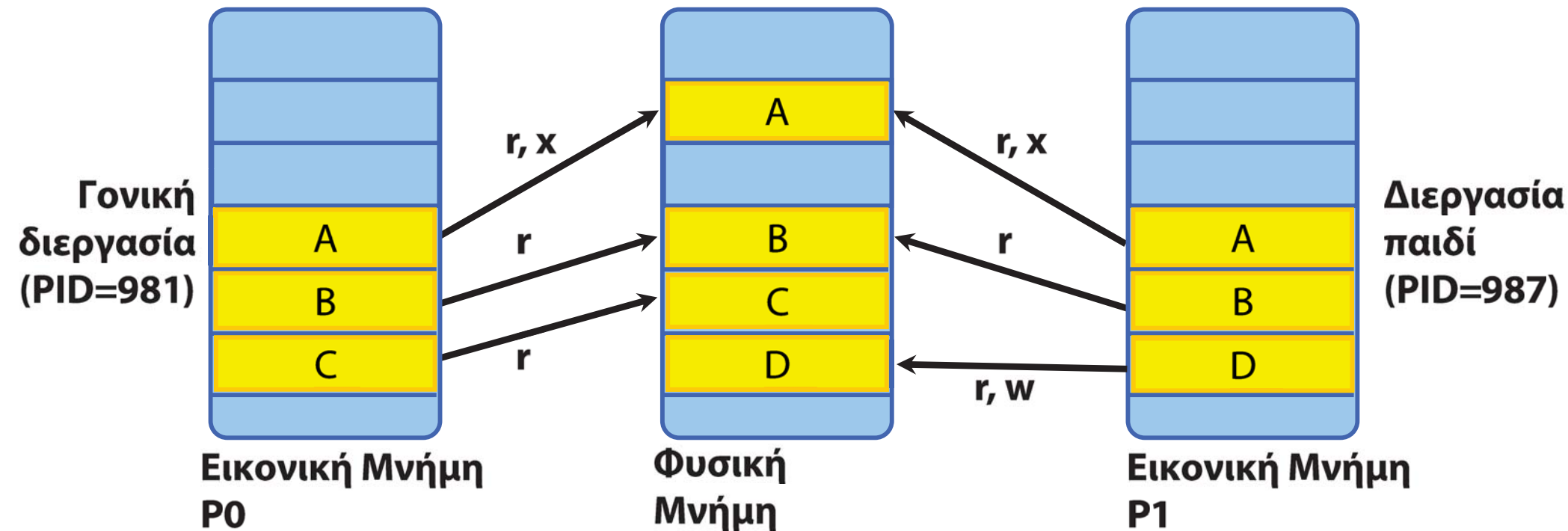
Copy-on-Write



Copy-on-Write

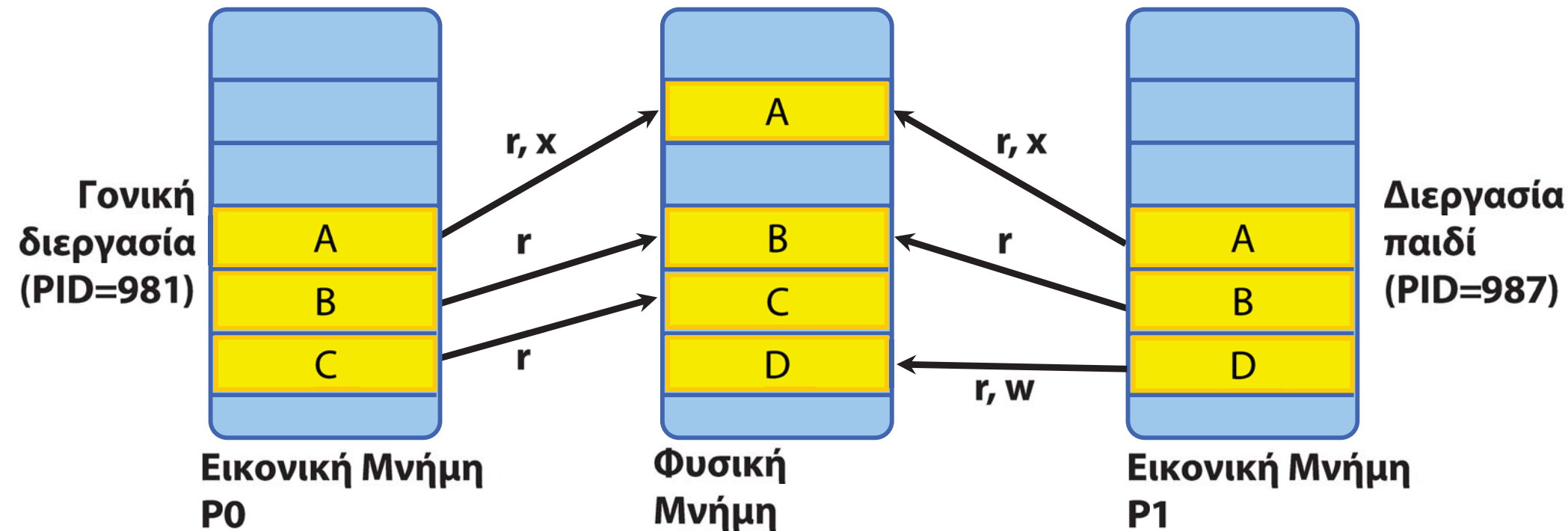


Copy-on-Write



- ◆ 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

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)



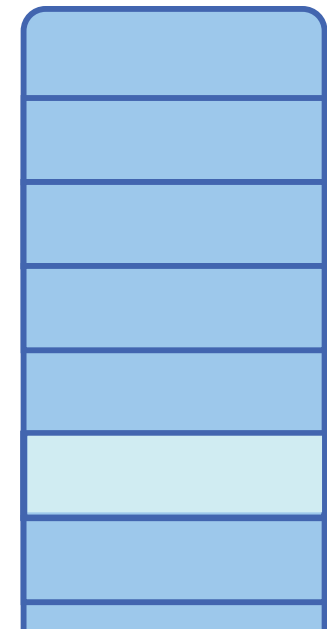
**Εικονική
Μνήμη P_0**

```
fd = open("file");  
vaddr = mmap(..., fd, ...);  
... access vaddr + offset C...
```

Απεικόνιση Αρχείων στη Μνήμη (2)



Εικονική
Μνήμη P_0



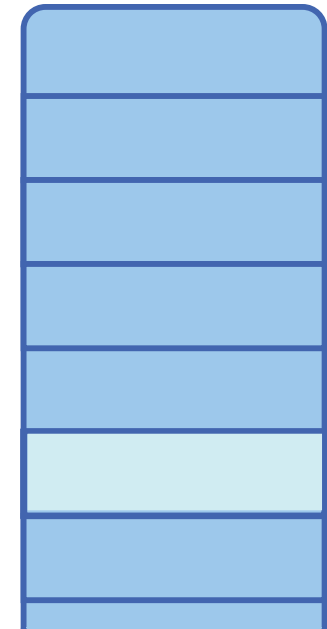
Φυσική Μνήμη

```
fd = open("file");  
vaddr = mmap(..., fd, ...);  
... access vaddr + offset C...
```

Απεικόνιση Αρχείων στη Μνήμη (2)



Εικονική
Μνήμη P_0



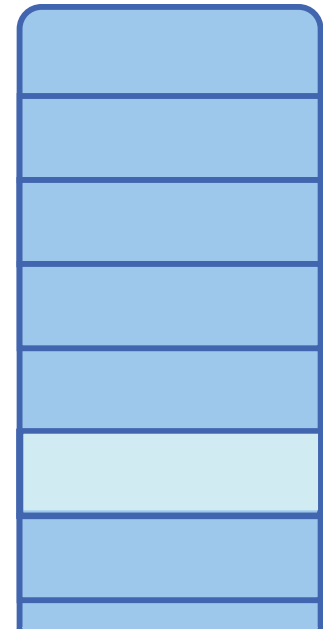
Φυσική Μνήμη

→ `fd = open("file");`
`vaddr = mmap(..., fd, ...);`
`... access vaddr + offset C...`

Απεικόνιση Αρχείων στη Μνήμη (2)

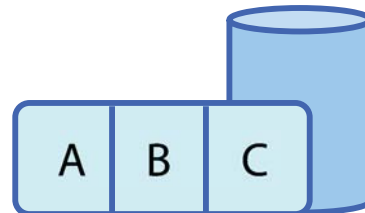


Εικονική
Μνήμη P_0



Φυσική Μνήμη

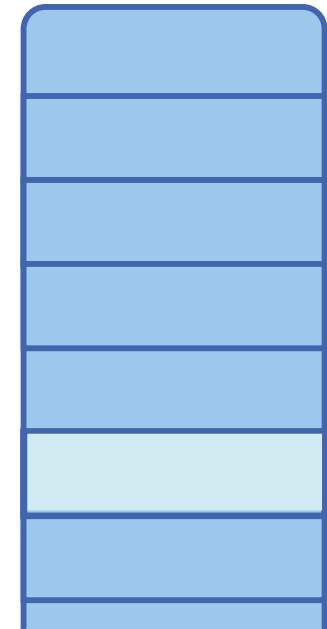
→ `fd = open("file");`
`vaddr = mmap(..., fd, ...);`
`... access vaddr + offset C...`



Απεικόνιση Αρχείων στη Μνήμη (2)

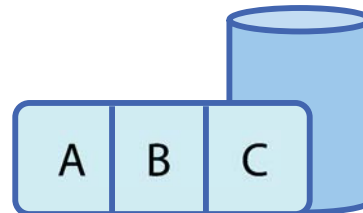


Εικονική
Μνήμη P_0

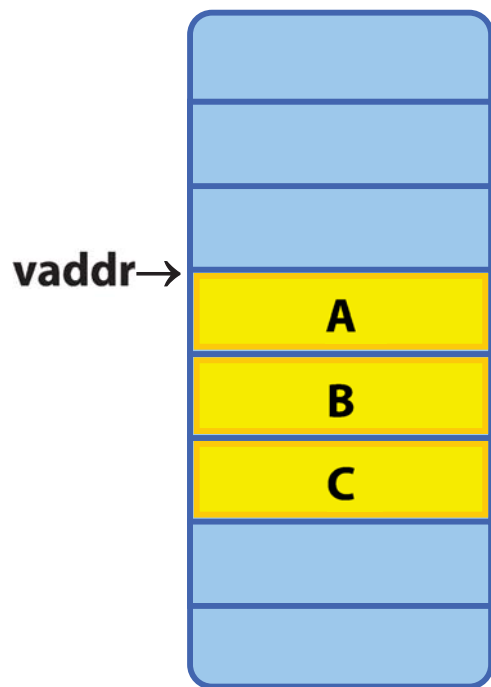


Φυσική Μνήμη

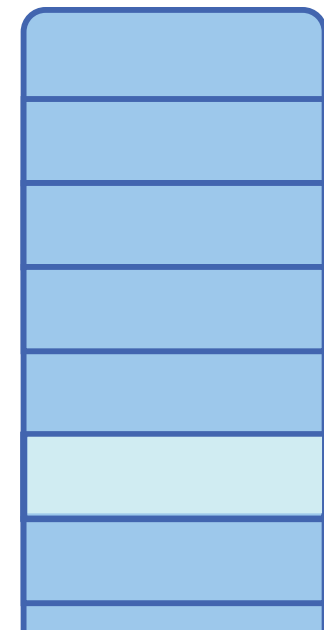
→ `fd = open("file");`
`vaddr = mmap(..., fd, ...);`
`... access vaddr + offset C...`



Απεικόνιση Αρχείων στη Μνήμη (2)

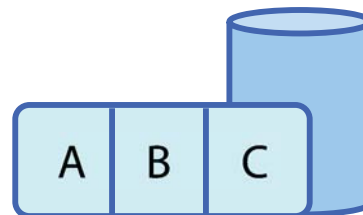


Εικονική
Μνήμη P_0

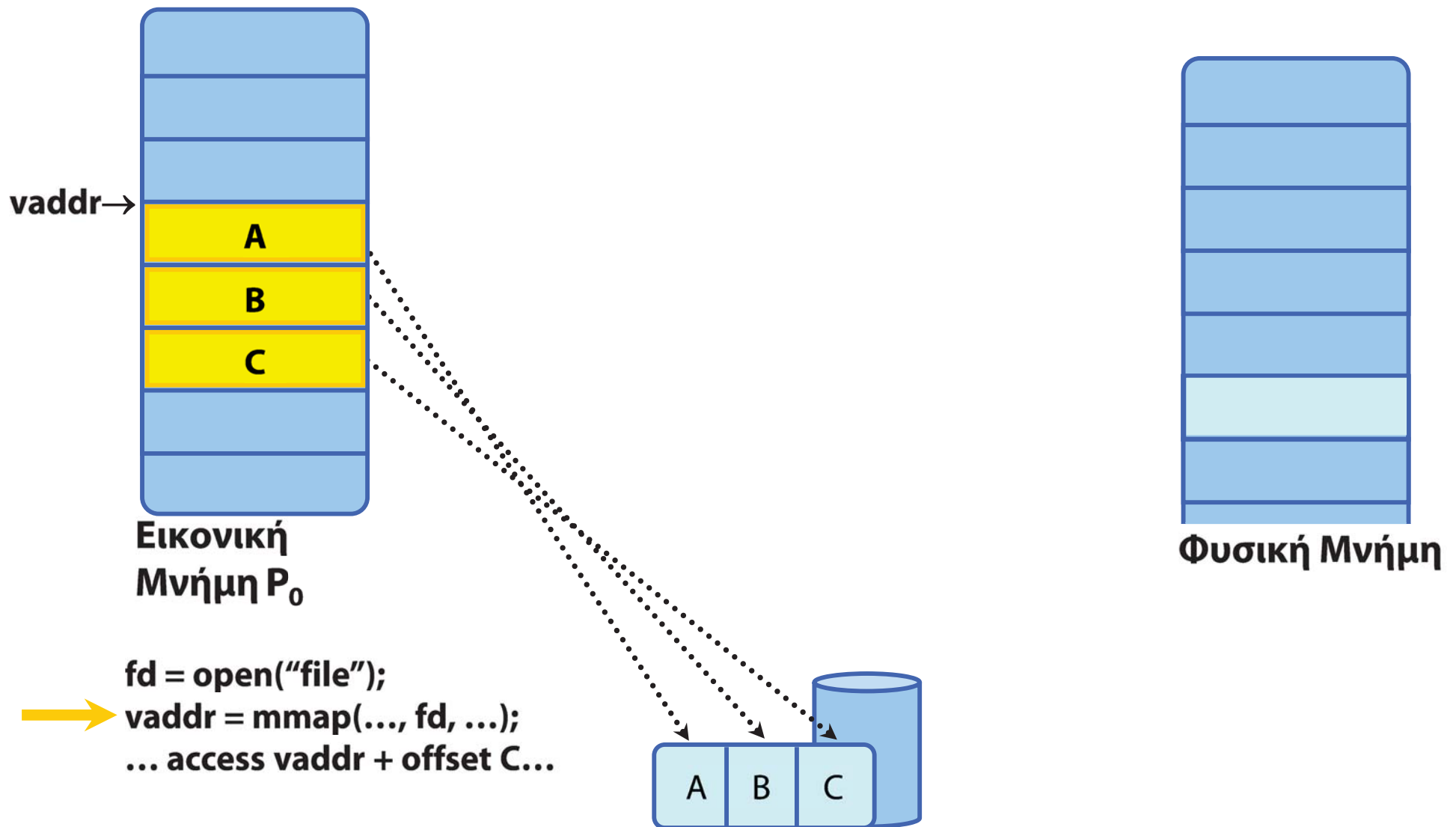


Φυσική Μνήμη

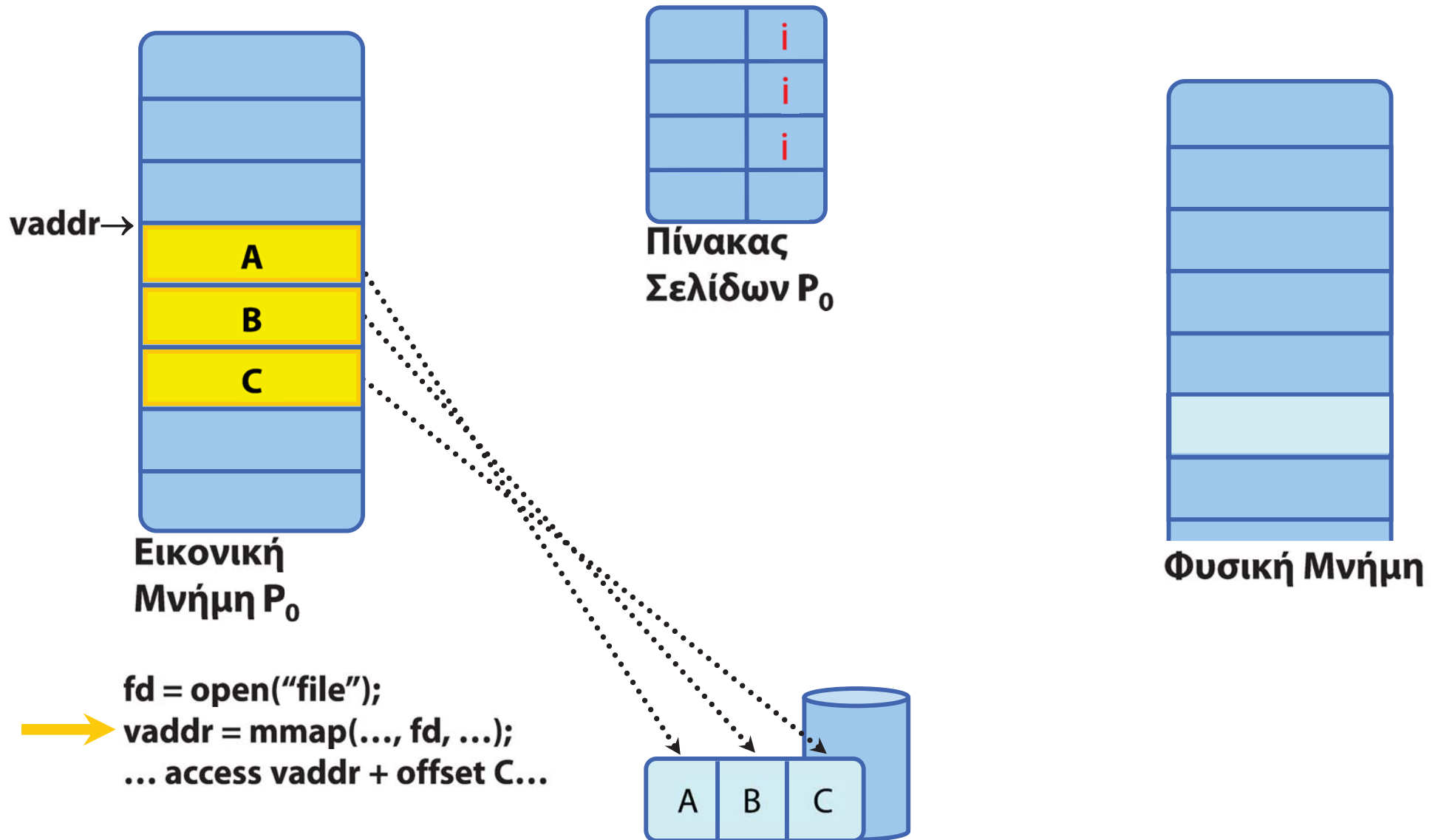
fd = open("file");
→ vaddr = mmap(..., fd, ...);
... access vaddr + offset C...



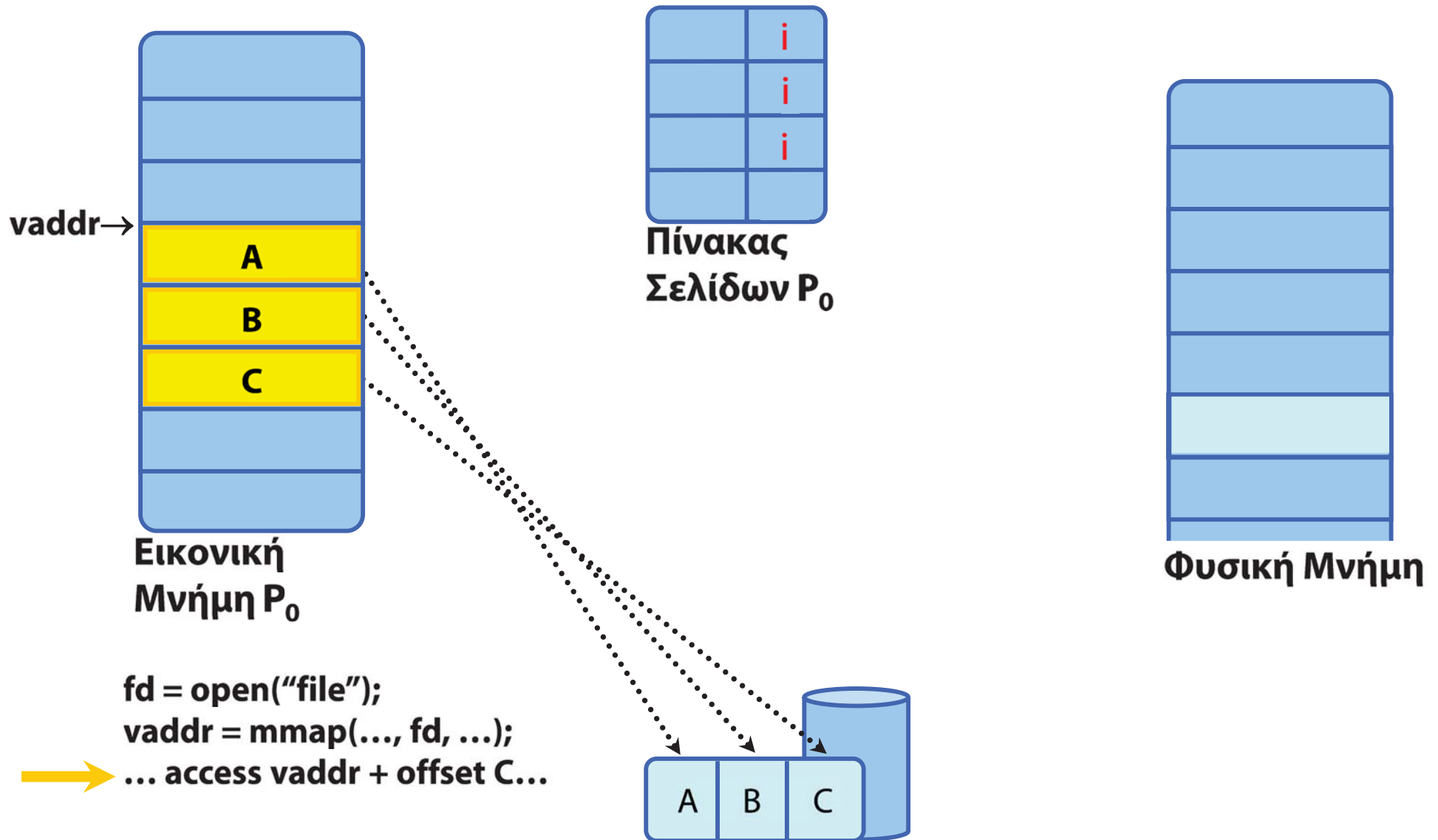
Απεικόνιση Αρχείων στη Μνήμη (2)



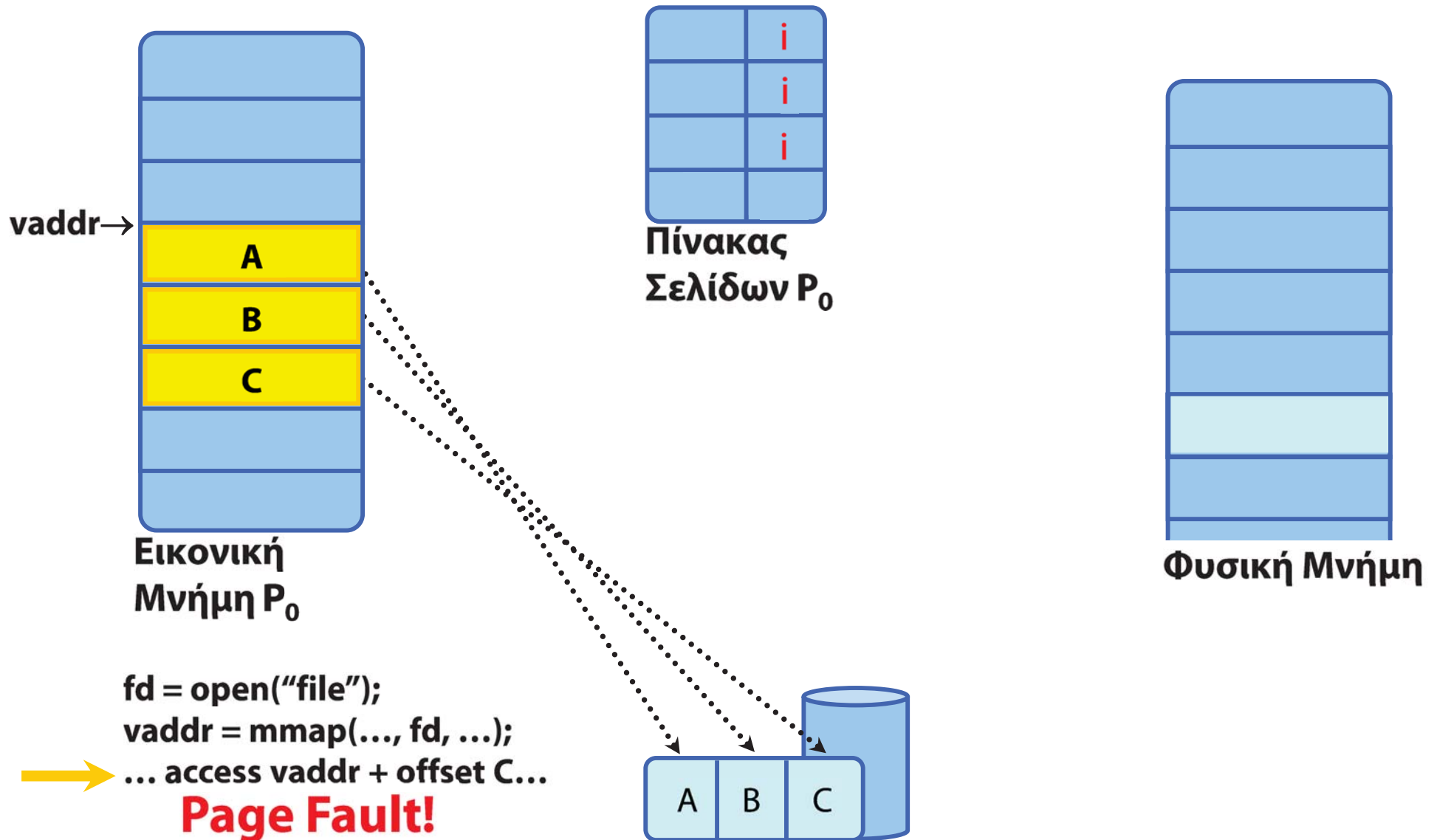
Απεικόνιση Αρχείων στη Μνήμη (2)



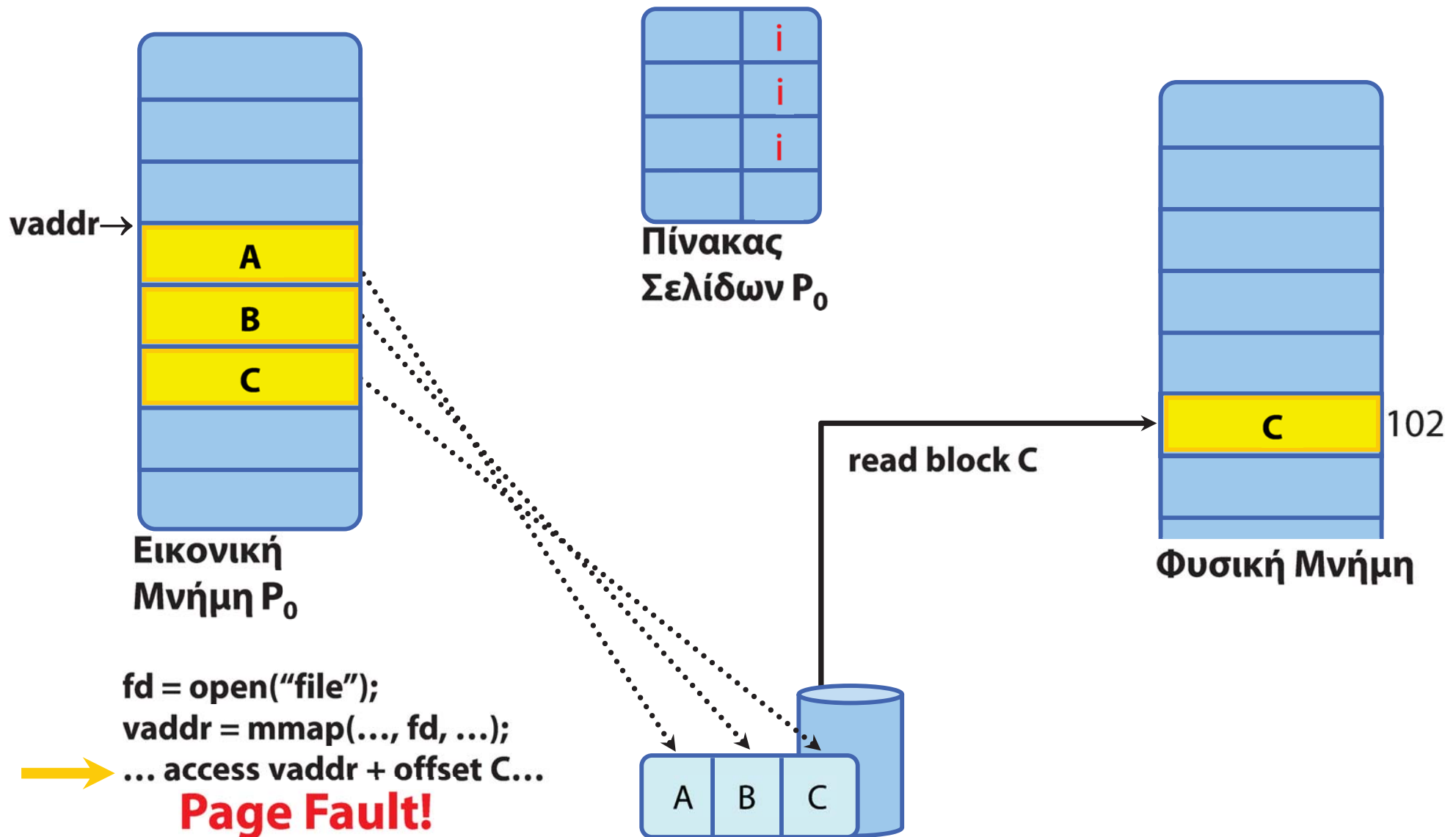
Απεικόνιση Αρχείων στη Μνήμη (2)



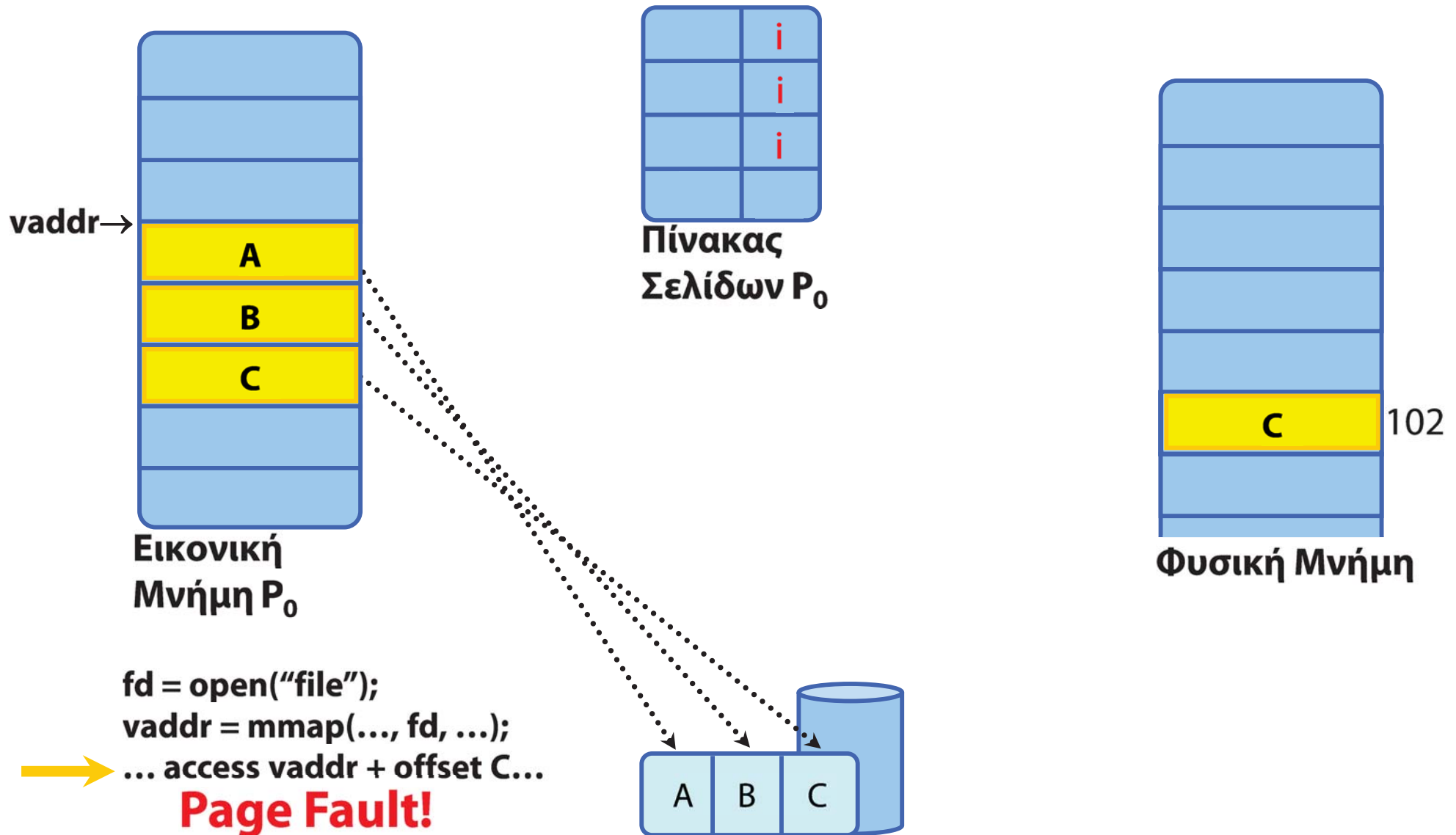
Απεικόνιση Αρχείων στη Μνήμη (2)



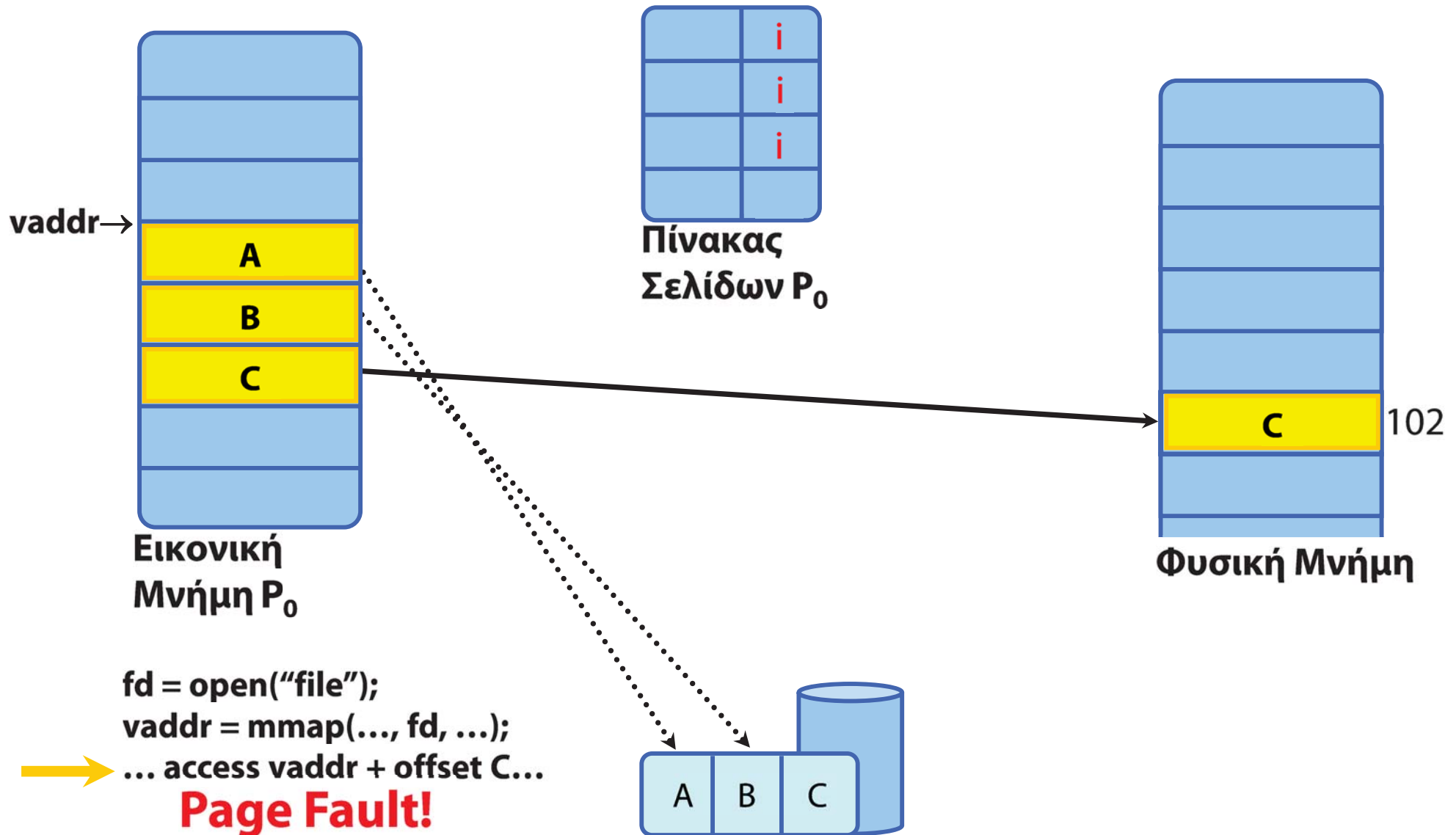
Απεικόνιση Αρχείων στη Μνήμη (2)



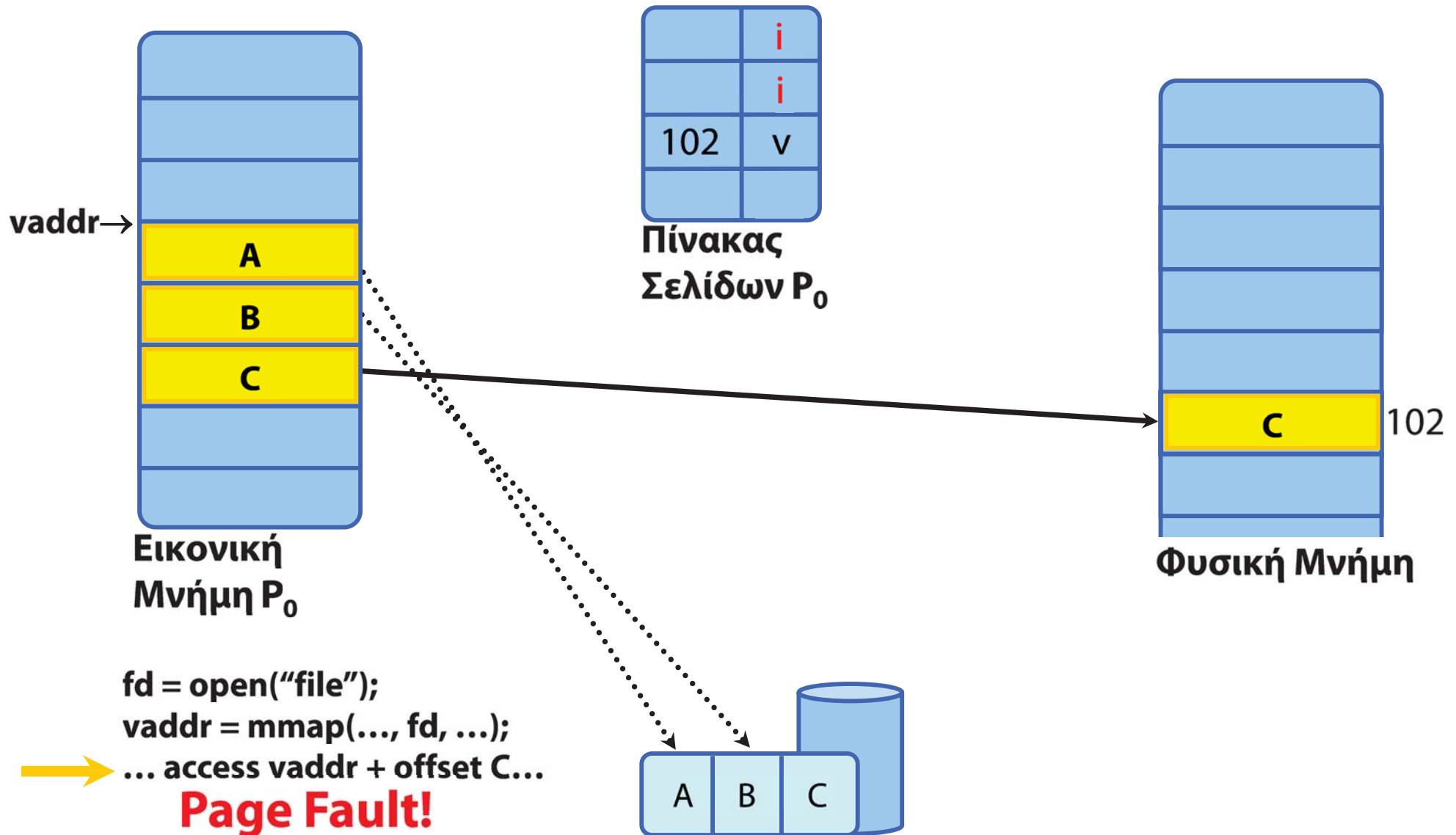
Απεικόνιση Αρχείων στη Μνήμη (2)



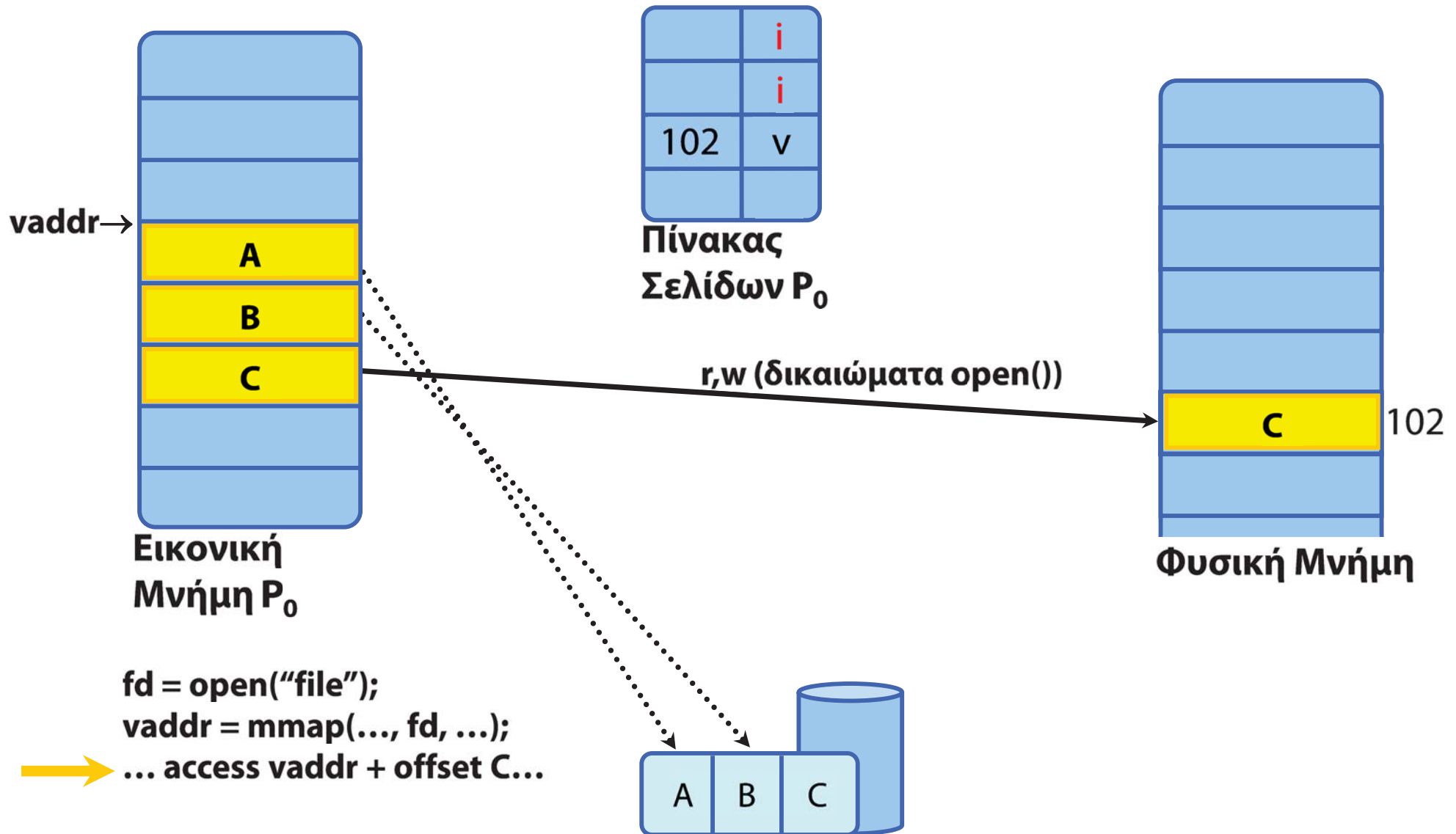
Απεικόνιση Αρχείων στη Μνήμη (2)



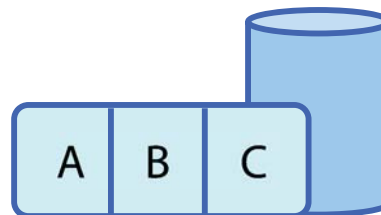
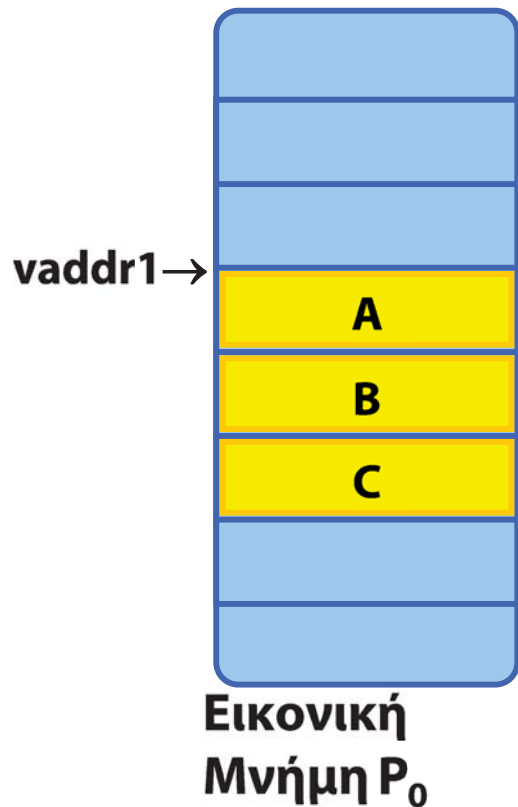
Απεικόνιση Αρχείων στη Μνήμη (2)



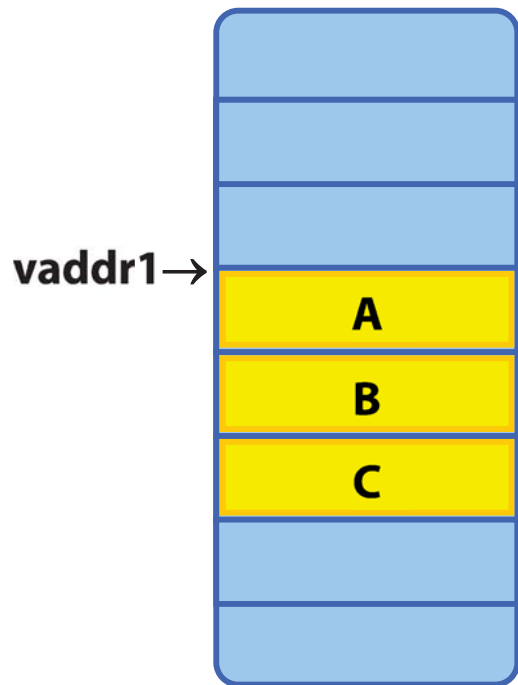
Απεικόνιση Αρχείων στη Μνήμη (2)



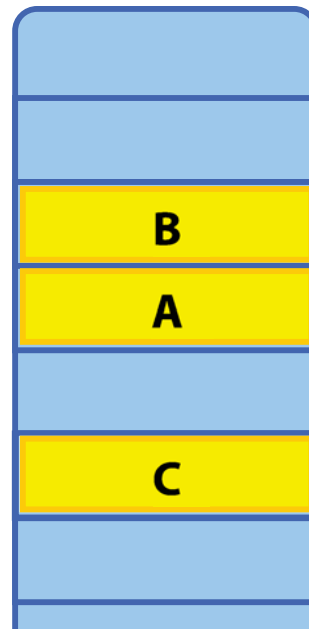
Απεικόνιση Αρχείων στη Μνήμη (2)



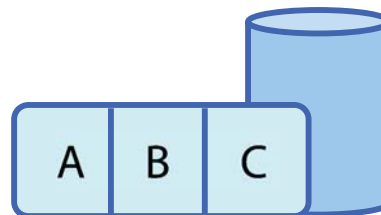
Απεικόνιση Αρχείων στη Μνήμη (2)



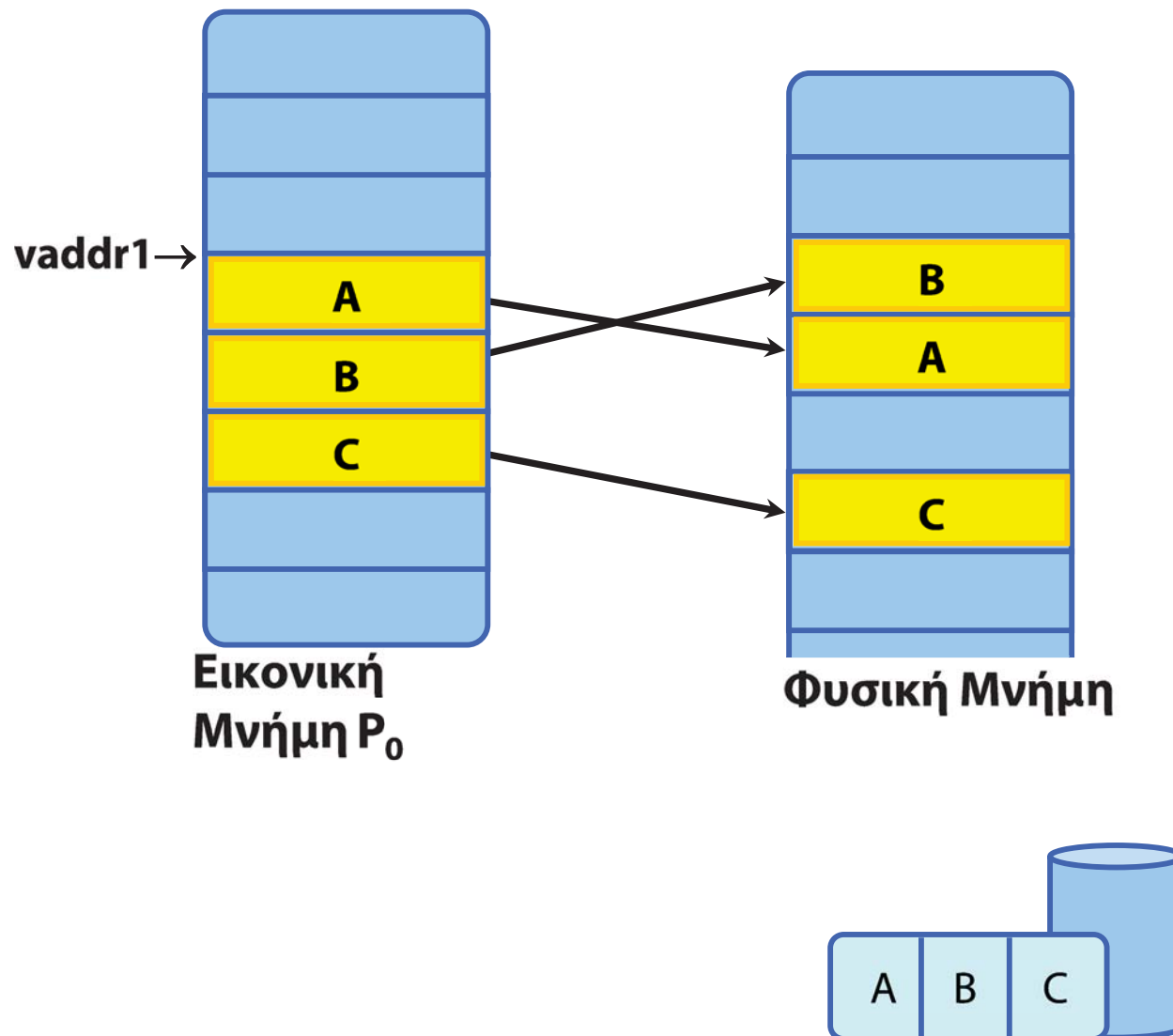
Εικονική
Μνήμη P_0



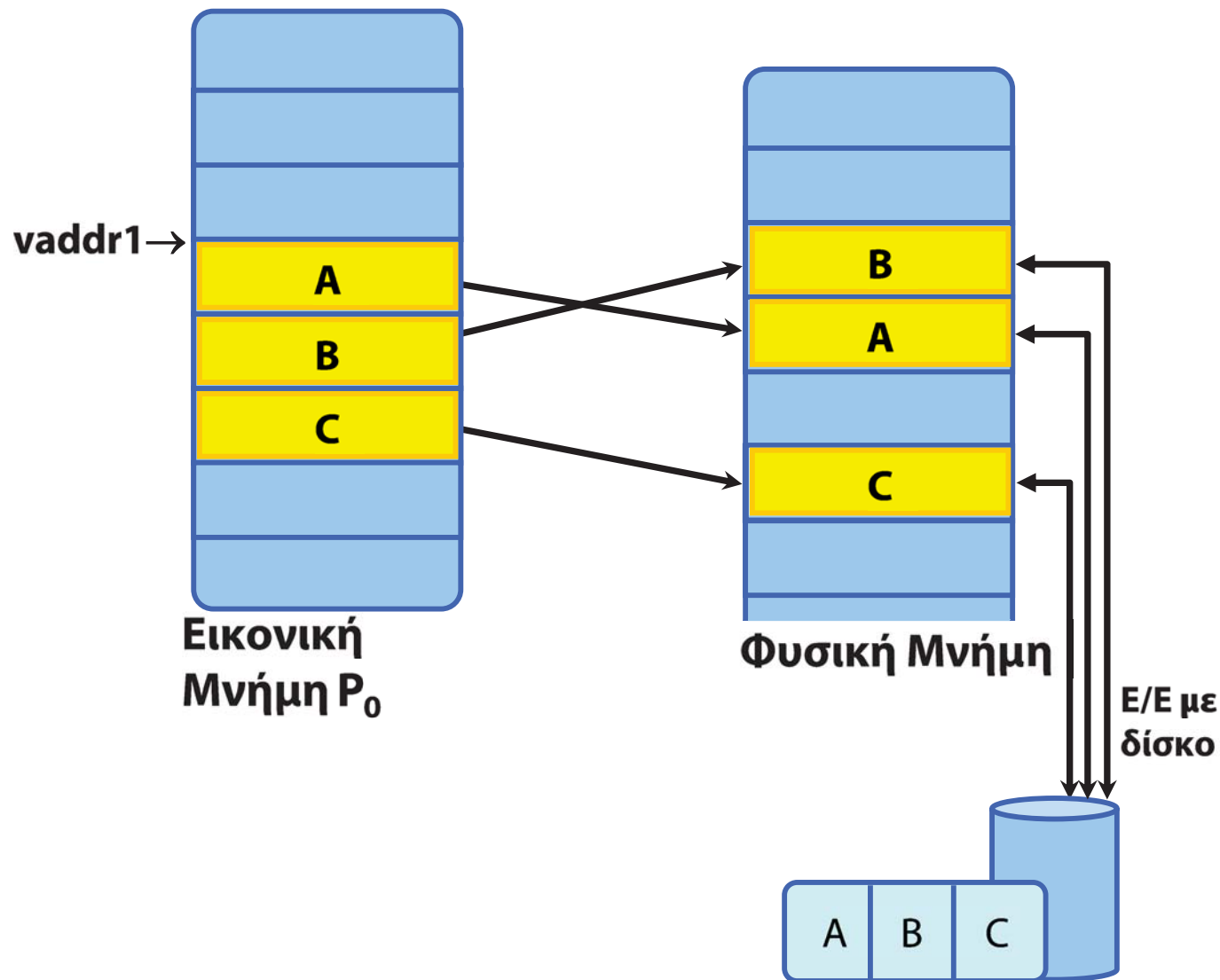
Φυσική Μνήμη



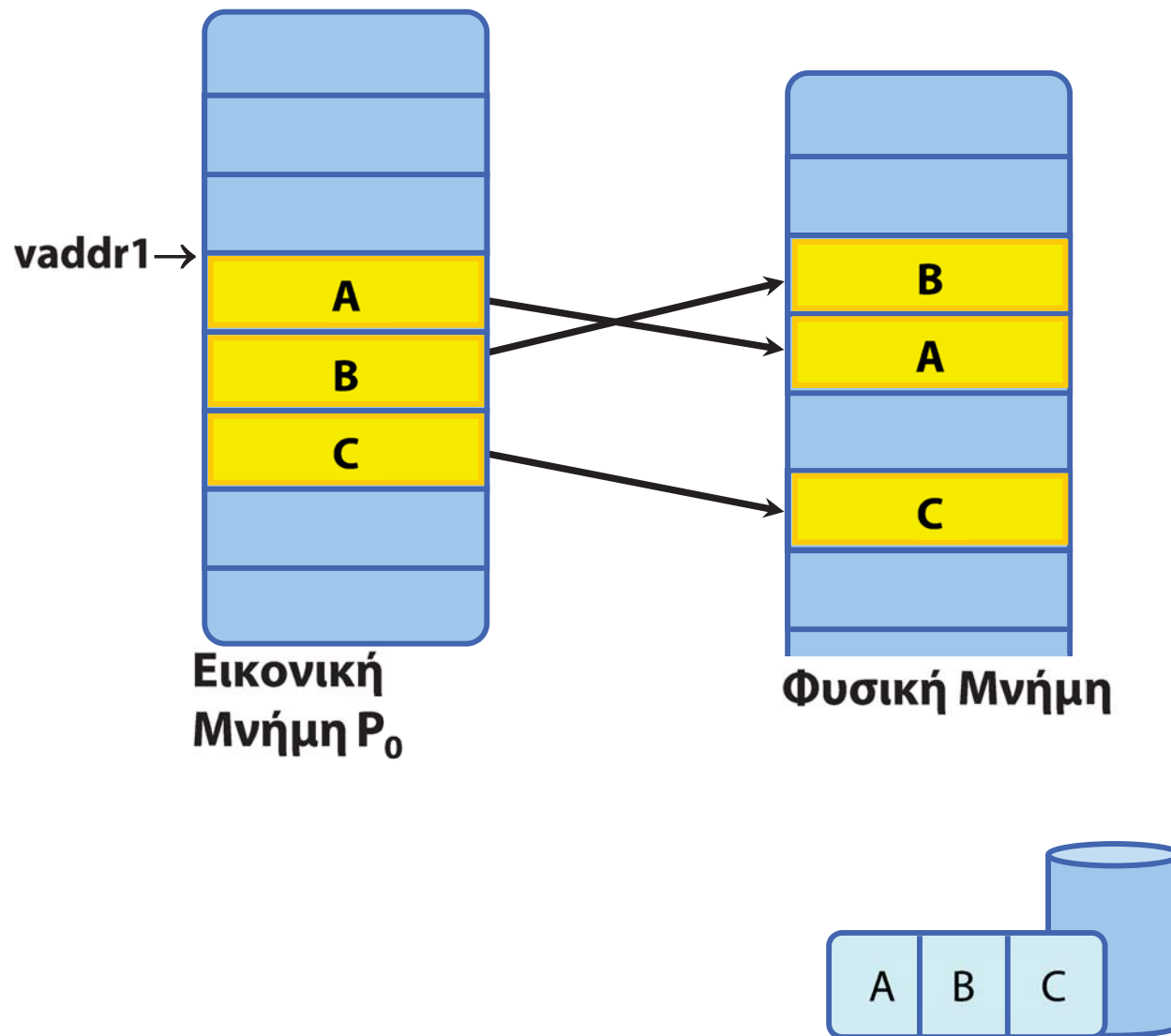
Απεικόνιση Αρχείων στη Μνήμη (2)



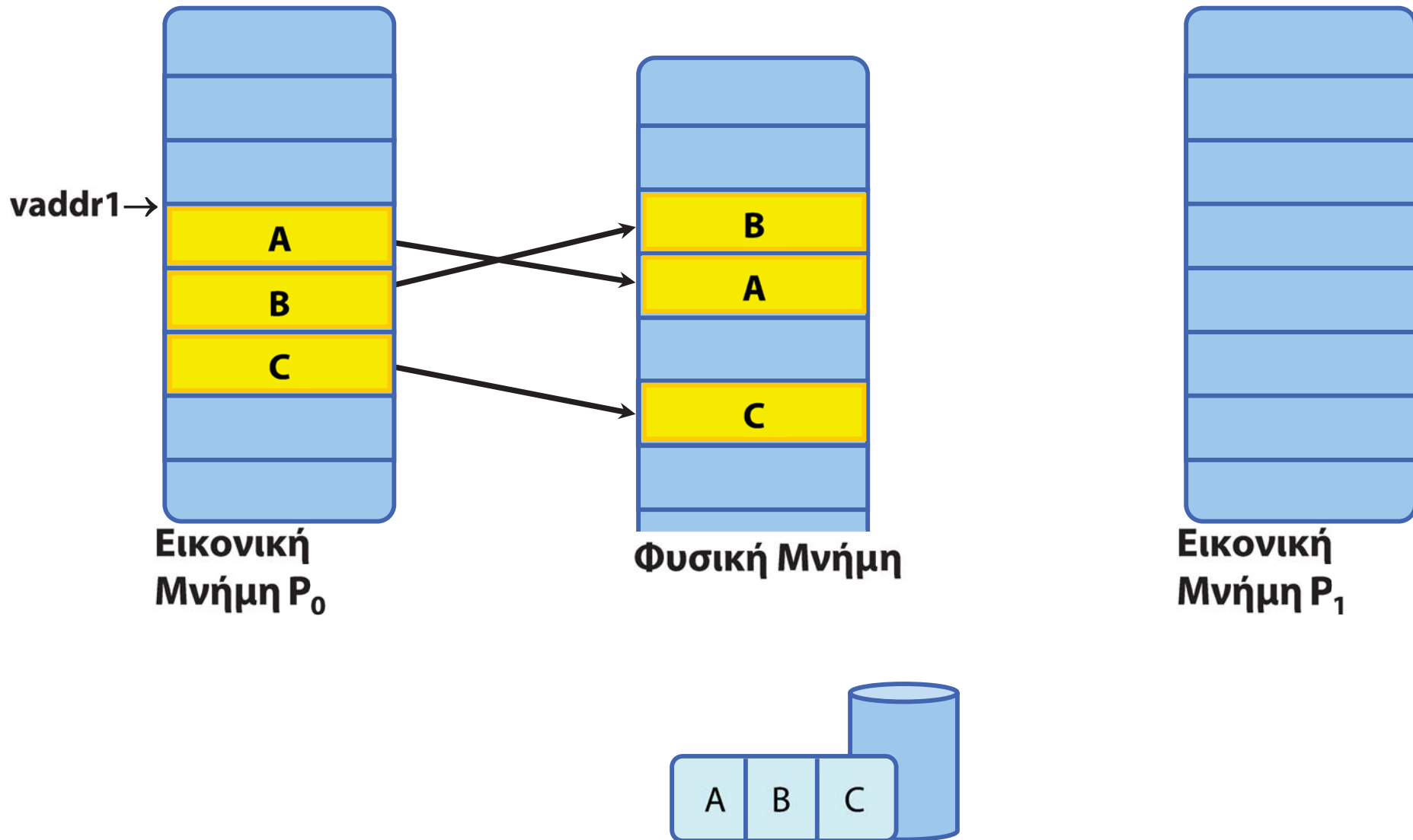
Απεικόνιση Αρχείων στη Μνήμη (2)



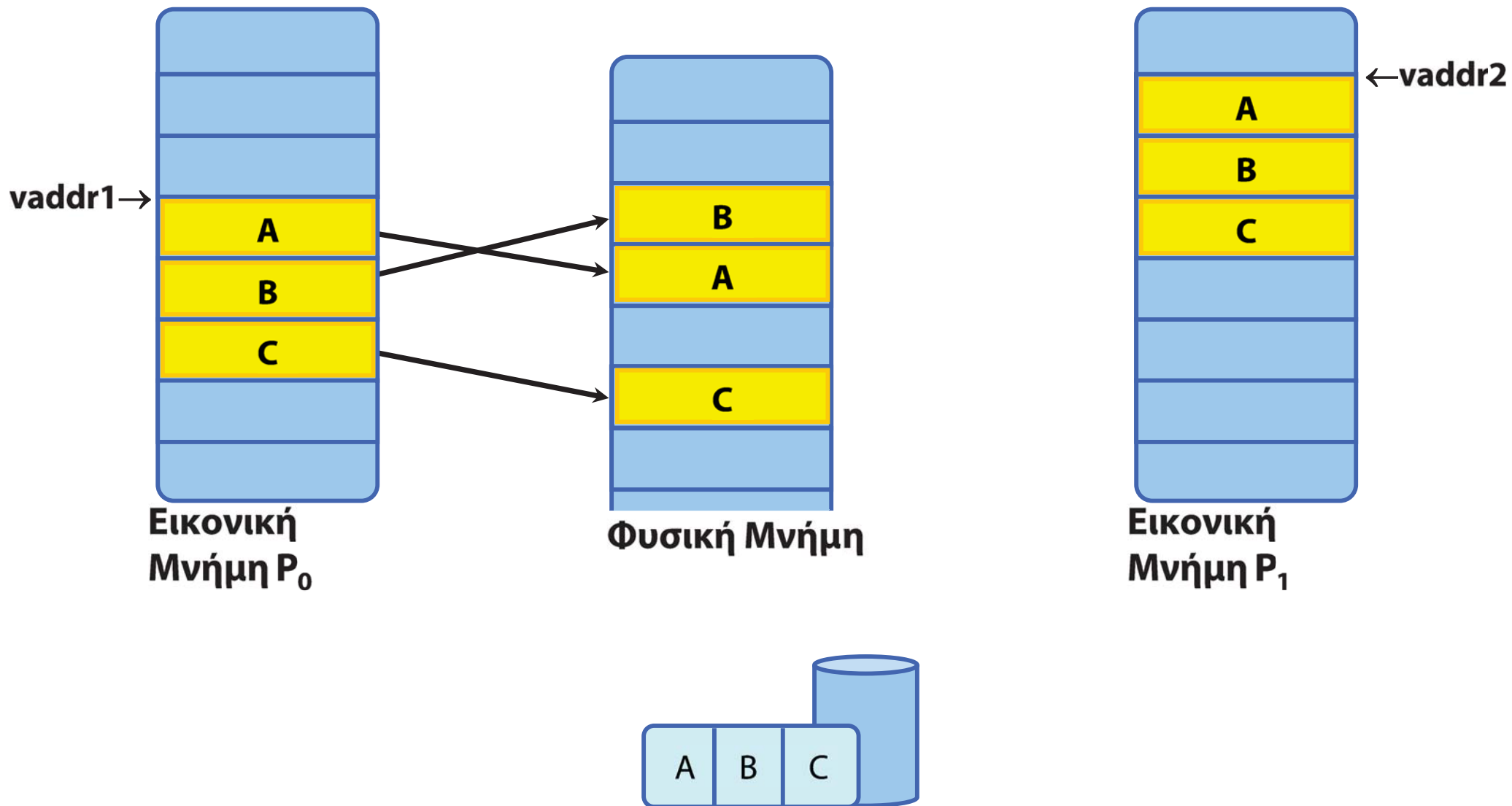
Απεικόνιση Αρχείων στη Μνήμη (2)



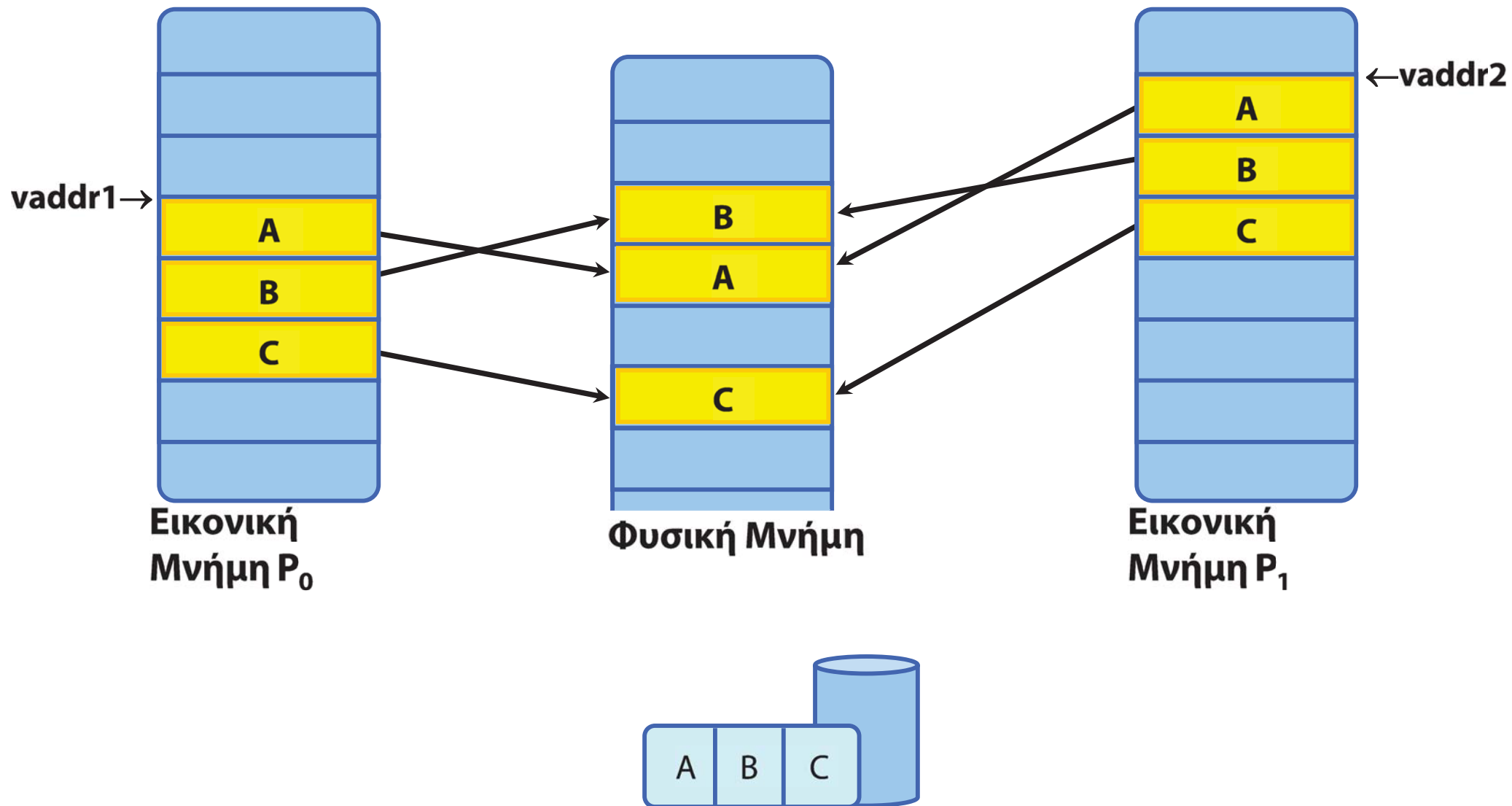
Απεικόνιση Αρχείων στη Μνήμη (2)



Απεικόνιση Αρχείων στη Μνήμη (2)



Απεικόνιση Αρχείων στη Μνήμη (2)



Ερωτήσεις;

