

ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ 3

ΔΙΑΧΕΙΡΙΣΗ ΜΝΗΜΗΣ

Πλάνο παρουσίασης

- Διαχείριση μνήμης
- Ιδεατή μνήμη

Διαχείριση μνήμης

- Κάθε πρόγραμμα που εκτελείται σε έναν υπολογιστή χρειάζεται μνήμη
 - Εντολές
 - Δεδομένα
 - Στοίβα

Διαχείριση μνήμης

- Πρέπει να υπάρχει διαχείριση της μνήμης
 - Υλικό ή/και λογισμικό που παραχωρεί τμήματα της μνήμης σε μια διαδικασία
- Αρχικά θα δούμε μερικά συστήματα διαχείρισης μνήμης που δεν χρησιμοποιούνται πια
 - Όμως από τα **μειονεκτήματά τους** θα καταλάβουμε πως φτάσαμε στα σύγχρονα συστήματα
 - Σελιδοποίηση, Τμηματοποίηση, Ιδεατή μνήμη

Σκέτη μηχανή

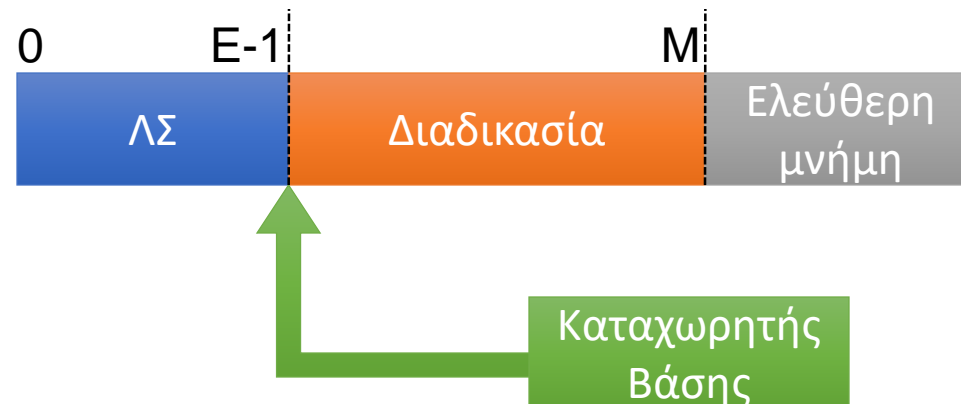
- Δεν υπάρχει καμία διαχείριση μνήμης
- Ο προγραμματιστής έχει πλήρη έλεγχο
- Μέγιστη ευελιξία
- Δεν παρέχει έτοιμες υπηρεσίες
 - Διαχείριση συσκευών
 - Κλήσεις συστήματος
 - Διαχείριση λαθών

Διαχειριστής μνήμης

- Ο Διαχειριστής Μνήμης (ΔΜ) (Memory Manager) πρέπει να παρέχει τις παρακάτω υπηρεσίες:
 - Να ξέρει ποια διαμερίσματα της Κύριας Μνήμης (ΚΜ) είναι κατειλημμένα
 - Να μπορεί να αναθέτει μνήμη σε διαδικασίες όταν την χρειάζονται
 - Να παίρνει πίσω μνήμη που “ελευθερώνεται” από διεργασίες
 - Να διαχειρίζεται την εναλλαγή (swapping)
 - Μεταφορά (διαμερισμάτων) διεργασιών μεταξύ ΚΜ & σκληρού δίσκου
- Οι ΔΜ χωρίζονται σε κατηγορίες ανάλογα με το αν επιτρέπουν σελιδοποίηση (paging) και εναλλαγή
 - Χωρίς αυτά ο ΔΜ είναι «απλός»

Μονοπρογραμματισμός (Monoprogramming)

- Μόνο μια διαδικασία βρίσκεται στη ΚΜ κάθε χρονική στιγμή
- Η ΚΜ χωρίζεται σε 2 μέρη
 - Ένα για το ΛΣ
 - Ένα για την διαδικασία που εκτελείται
- Μόλις ο χρήστης ζητήσει να τρέξει ένα πρόγραμμα, το ΛΣ το "φορτώνει" στη ΚΜ και το εκτελεί

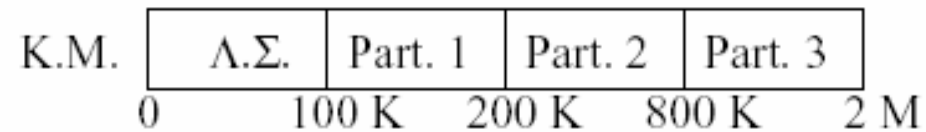


Μονοπρογραμματισμός (Monoprogramming)

- Ο μονοπρογραμματισμός δεν είναι καλή λύση για τα περισσότερα υπολογιστικά συστήματα
 - Εφαρμογές με μεγάλο ποσοστό λειτουργιών Εισόδου/Εξόδου
 - Η CPU υποαπασχολείται
 - Διαδραστικές εφαρμογές
 - Ο χρόνος απόκρισης είναι πολύ μεγάλος
 - Πολλές εφαρμογές θέλουν την δυνατότητα δημιουργίας πολλών διεργασιών που τρέχουν “ταυτόχρονα”
- Χρειαζόμαστε πολυπρογραμματισμό (multiprogramming)
 - Η δουλειά του ΔΜ γίνεται πολύ πιο δύσκολη!

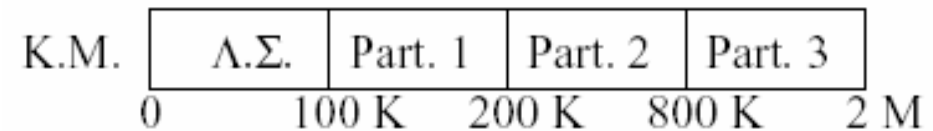
ΔΜ με Σταθερές Περιοχές (Fixed Regions)

- Η μνήμη χωρίζεται σε περιοχές
- Κάθε περιοχή είναι σταθερού μεγέθους. Διαφορετικές περιοχές μπορεί να έχουν διαφορετικό μέγεθος.
- Σε κάθε περιοχή μπορεί να φορτωθεί μια διαδικασία **ολόκληρη** (το μέγεθος της οποίας πρέπει να είναι \leq του μεγέθους της περιοχής)



- Ο Δ.Μ. διατηρεί μια ουρά από διαδικασίες που περιμένουν να εκτελεστούν. Μπορεί να υπάρχει μια ουρά για όλες τις περιοχές ή μια ουρά για κάθε περιοχή.
 - Στην 1^η περίπτωση ο ΔΜ πρέπει να προσέξει να μη σπαταλάει μνήμη
 - Στην 2^η περίπτωση μπορεί να υπάρχει διαθέσιμη μνήμη και να μη χρησιμοποιείται

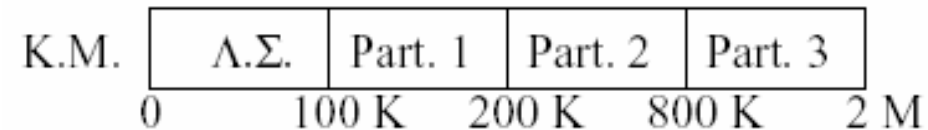
ΔΜ με Σταθερές Περιοχές (Fixed Regions)



- Αν μια μικρή διαδικασία μπει στην περιοχή 3, τότε αν και υπάρχει πολύ διαθέσιμος χώρος σε αυτό το διαμέρισμα δεν μπορεί να χρησιμοποιηθεί. Αυτό ονομάζεται **εσωτερικός κατακερματισμός** (internal fragmentation).
- Η λύση στο πρόβλημα δεν είναι προφανής
 - Αν κάθε φορά ο Δ.Μ. ψάχνει για τη διαδικασία στην ουρά που ελαχιστοποιεί τον εσωτερικό κατακερματισμό, τότε μικρές διεργασίες θα αδικούνται (λιμοκτονούν)

Θεμελιώδη Προβλήματα των Σταθερών Περιοχών

- Ακόμα και με μονοπρογραμματισμό υπάρχουν 2 διαμερίσματα
 - Ένα για το ΛΣ και ένα για την εφαρμογή



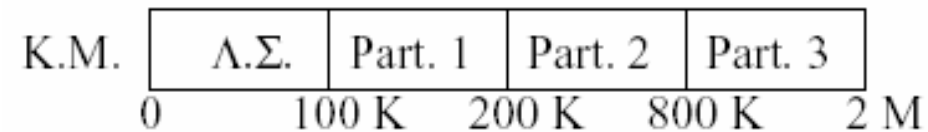
- Η πρώτη διεύθυνση μνήμης της διαδικασίας είναι η 0
 - Στη φυσική διεύθυνση 0 βρίσκονται εντολές ή δεδομένα του ΛΣ
- Η λύση εστιάζει στη μετάφραση των «λογικών» διευθύνσεων σε «φυσικές»
- Για παράδειγμα, αν το process έχει μπει στην Περιοχή 1 τότε σε όλες τις διευθύνσεις που παράγονται κατά την εκτέλεση του process πρέπει να προστίθεται το 100K
 - Πρέπει να υπάρχει κατάλληλο υλικό

Θεμελιώδη Προβλήματα των Σταθερών Περιοχών

- Συνήθως η παραπάνω στρατηγική υλοποιείται με την χρησιμοποίηση ενός ειδικού καταχωρητή βάσης (base register).
- Στο παραπάνω παράδειγμα, όταν η διαδικασία φορτωθεί στην Περιοχή 1
 - Αμέσως πριν δοθεί η CPU στην διαδικασία ο καταχωρητής βάσης παίρνει την τιμή 100K
 - Αυτό γίνεται από το ΛΣ
 - Οι διεργασίες δεν έχουν πρόσβαση σε αυτόν τον καταχωρητή

Θεμελιώδη Προβλήματα των Σταθερών Περιοχών

- Στην περίπτωση πολυπρογραμματισμού, πρέπει το σύστημα να εγγυάται ότι κάθε πρόγραμμα θα προστατεύεται από τα άλλα που είναι στη ΚΜ



- Απαιτείται έλεγχος σχετικά με τις διευθύνσεις που μπορεί να παράγει μια διαδικασία
 - Η διαδικασία στην Περιοχή 1 δεν μπορεί να παράγει διευθύνσεις > 200K
 - Θα επηρεάσει την διαδικασία που υπάρχει στην Περιοχή 2

Θεμελιώδη Προβλήματα των Σταθερών Περιοχών

- Αυτό το πρόβλημα λύνεται με χρήση ενός ακόμα ειδικού καταχωρητή
 - Καταχωρητής ορίου (limit register)
- Ο καταχωρητής ορίου περιέχει την μέγιστη διεύθυνση της περιοχής την οποία χρησιμοποιεί μια διαδικασία
- Κάθε διεύθυνση που παράγεται
 - Προστίθεται στον καταχωρητή βάσης
 - Συγκρίνεται με τον καταχωρητή ορίου
 - Αν το αποτέλεσμα της άθροισης είναι μεγαλύτερο από το περιεχόμενο του καταχωρητή ορίου
 - Μη έγκυρη διεύθυνση μνήμης
 - Η διαδικασία τερματίζεται

Θεμελιώδη Προβλήματα των Σταθερών Περιοχών

- Με την χρήση καταχωρητών βάσης και ορίου οι διαδικασίες μπορούν να μετακινηθούν στην ΚΜ
 - Ανάθεση σε διαφορετικές περιοχές κατά την διάρκεια της εκτέλεσής τους
- Αυτό ονομάζεται **επανατοποθέτηση** (relocation)

ΔΜ με Εναλλαγή (Swapping)

- Μπορεί να υποβληθούν προς εκτέλεση περισσότερες διαδικασίες από όσες μπορεί να υποστηρίξει η ΚΜ
 - Ορίζεται ένας χώρος στην επικουρική μνήμη (π.χ. στον σκληρό δίσκο) που φιλοξενεί τις διεργασίες που δεν «χωρούν» στη ΚΜ
- Η μεταφορά διεργασιών μεταξύ ΚΜ και επικουρικής μνήμης στο δίσκο ονομάζεται **εναλλαγή** (swapping)

ΔΜ με Μεταβλητές Διαιρέσεις

- Οι σταθερές περιοχές έχουν σημαντικό θέμα
 - Δύσκολο να γίνει καλό ταίριασμα μεταξύ μεγεθών περιοχών και διαδικασιών
 - Σπαταλιέται πολύς χώρος ΚΜ
- Μεταβλητές Διαιρέσεις (variable partitions)

ΔΜ με Μεταβλητές Διαιρέσεις

- Οι διευθύνσεις, το πλήθος και τα μεγέθη των περιοχών μεταβάλλονται ανάλογα με τις ανάγκες
- Κάθε διαμέρισμα αποτελείται από συνεχόμενες διευθύνσεις στην ΚΜ και μπορεί να είναι ίσου μεγέθους με τη διαδικασία που το χρησιμοποιεί
 - Τα μεταβαλλόμενα partitions ελαχιστοποιούν την σπατάλη χώρου στην ΚΜ
 - Η ανάθεση ΚΜ σε διαδικασίες και το έργο διαχείρισης της ΚΜ δυσκολεύει

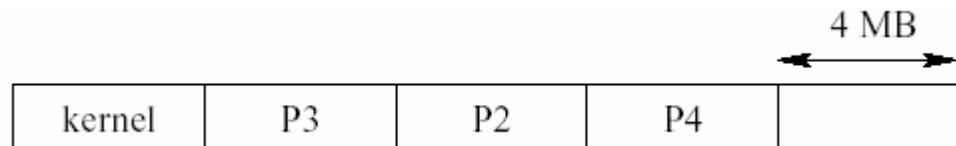
ΔΜ με Μεταβλητές Διαιρέσεις

- Παράδειγμα:
 - Υπάρχουν τμήματα της ΚΜ που μένουν άδεια
 - Στο παράδειγμα μας έστω με μέγεθος 3ΜΒ και 1ΜΒ

ΚΜ	Kernel	P1			Μόλις ήρθε το process P1
>>	Kernel	P1	P2		Μόλις ήρθε και το process P2
>>	Kernel		P2		Έφυγε το process P1
>>	Kernel	P3	P2		Ήρθε το process P3
>>	Kernel	P3	P2	P4	Ήρθε το process P4

ΔΜ με Μεταβλητές Διαιρέσεις

- Αν έρθει μια διαδικασία P5 που χρειάζεται 4MB
 - Δεν μπορεί να εκτελεστεί
 - Υπάρχει 4MB ελεύθερος χώρος στη ΚΜ αλλά όχι συνεχόμενος
- **Εξωτερικός κατακερματισμός (external fragmentation)**
- Πιθανή λύση
 - Σύμπτυξη μνήμης (storage compaction)
 - Ενώνει όλες τις ελεύθερες περιοχές στην ΚΜ σε ένα συνεχόμενο διαμέρισμα της ΚΜ



- Χρησιμοποιείται σπάνια
 - Απαιτεί αντιγραφή μεγάλων περιοχών μνήμης και είναι χρονοβόρο

Σύγχρονες τεχνικές διαχείρισης μνήμης

- Σε σύγχρονα συστήματα οι δύο παρακάτω προσεγγίσεις χρησιμοποιούνται ως επί το πλείστον
 - Σελιδοποίηση (paging)
 - Οδηγεί εύκολα στην έννοια της **ιδεατής μνήμης (virtual memory)**
 - Τμηματοποίηση
- Επίσης οι συνδυασμοί των παραπάνω
 - Σελιδοποιημένη τμηματοποίηση
 - Τμηματοποιημένη σελιδοποίηση
- Βασικό χαρακτηριστικό των παραπάνω σύγχρονων τεχνικών διαχείρισης μνήμης είναι η εκχώρηση μη-συνεχόμεμων περιοχών κεντρικής μνήμης στις διαδικασίες (non-contiguous allocation)
 - Παρόλο που η φυσική μνήμη που δίνεται σε μία διαδικασία είναι μη-συνεχόμενη το ΛΣ δίνει την ψευδαίσθηση στον χρήστη ότι έχει στη διάθεσή του ολόκληρη τη μνήμη και ότι οι περιοχές που δίνονται σε αυτήν είναι συνεχείς/ενιαίες.
 - Η ψευδαίσθηση αυτή είναι που οδηγεί στην έννοια της ιδεατής μνήμης.

Σύγχρονες τεχνικές διαχείρισης μνήμης

- Στις σύγχρονες αυτές τεχνικές, μία διαδικασία **σπανίως βρίσκεται ολόκληρη** στην κεντρική μνήμη.
 - Μπορεί να μην χωράει εξαιτίας πολυπρογραμματισμού
 - Συνήθως βρίσκεται μόνο ένα μικρό τμήμα αυτής στην κεντρική μνήμη, εκείνο που απαιτείται αυτην την στιγμή για την εκτέλεση της διαδικασίας
 - Το υπόλοιπο τμήμα βρίσκεται στην δευτερεύουσα μνήμη

Σελιδοποίηση (Paging)

- Οι **φυσικές διευθύνσεις μνήμης (δλδ η φυσική μνήμη)** χωρίζονται σε μικρότερες συνεχείς μονάδες
 - Οι μονάδες αυτές καλούνται **Πλαίσια σελίδων (page frames)**
- Ο **χώρος διευθύνσεων κάθε διαδικασίας** χωρίζεται σε μικρότερες συνεχείς μονάδες
 - Οι μονάδες αυτές καλούνται **Σελίδες (pages)**
 - Χώρος διευθύνσεων: οι διεύθυνσεις που παράγονται από την CPU κατά την εκτέλεση μιας διαδικασίας και προσδιορίζουν τη θέση των διάφορων στοιχείων αυτής (όπως εντολές, μεταβλητές, τιμές κλπ) από τη σκοπιά της εκτελούμενης διαδικασίας.
- Κάθε διαδικασία θεωρεί πως έχει όλη τη μνήμη δική της και πως αυτή είναι συνεχής
- Στην σελιδοποίηση **Μέγεθος σελίδας == Μέγεθος πλαισίου σελίδας**
 - Τυπικά από 512 bytes έως 8 Kbytes

Σελιδοποίηση (Paging)

- Κατά την εκτέλεση μιας διαδικασίας, το υλικό μαζί με το ΛΣ τοποθετεί την απαιτούμενη σελίδα μνήμης διαδικασίας σε ένα πλαίσιο μνήμης
 - Οι σελίδες μιας διαδικασίας «μπαίνουν» σε πλαίσια μνήμης για να εκτελεστούν
 - Σελίδες μπορούν να τοποθετηθούν σε οποιαδήποτε ελεύθερο πλαίσιο μνήμης
 - Τα πλαίσια που καταλαμβάνουν οι σελίδες μιας διαδικασίας δεν χρειάζεται να είναι γειτονικά. Μπορεί να είναι οπουδήποτε εντός της φυσικής μνήμης.
 - Το ΛΣ διατηρεί έναν **“Πίνακα σελίδων” (Page table)** για να γνωρίζει ποια σελίδα μιας διαδικασίας έχει τοποθετηθεί σε ποιο πλαίσιο σελίδας της φυσικής μνήμης
 - Κάθε διαδικασία διατηρεί τον **δικό της πίνακα σελίδων**.

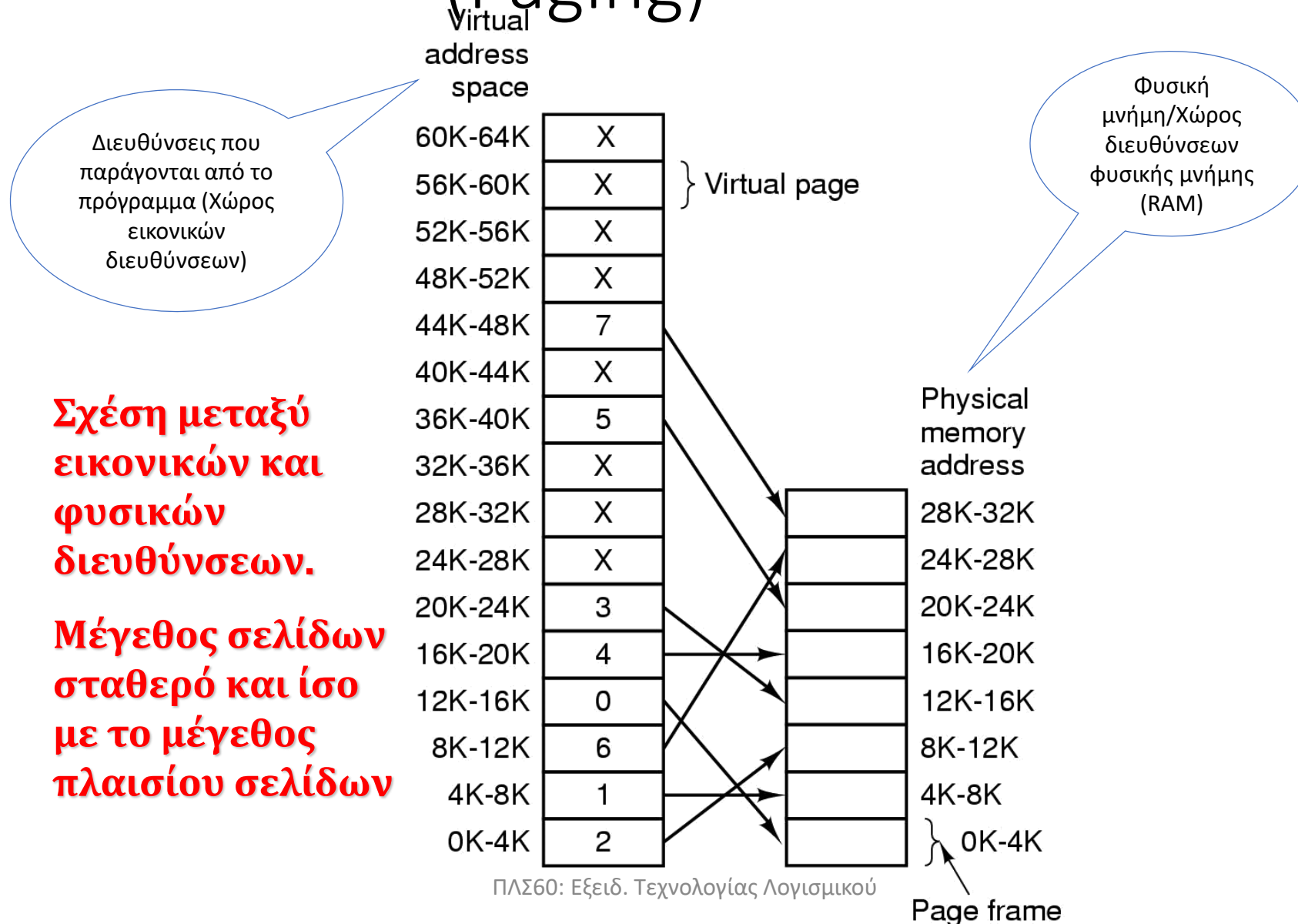
Σελιδοποίηση (Paging)

- Το σύνολο των διευθύνσεων μνήμης που παράγονται κατά την εκτέλεση ενός προγράμματος ορίζει τον χώρο διευθύνσεων (address space) της διαδικασίας.
 - Στη γενική περίπτωση, αυτό μπορεί να είναι μεγαλύτερο της κεντρικής μνήμης (KM)
 - Σε ένα 64-bit σύστημα μπορούμε να διευθυνσιοδοτήσουμε $2^{64} = 16$ exabytes (kilo → mega → giga → tera → peta → exa) θέσεις
 - Κανένα σύστημα δεν έχει τόσο μνήμη
- Στη σελιδοποίηση, οι διευθύνσεις που παράγονται από ένα πρόγραμμα λέγονται **εικονικές διευθύνσεις ή ιδεατές διευθύνσεις** (virtual addresses) και ορίζουν τον **εικονικό ή λογικό χώρο διευθύνσεων** της διαδικασίας
- Καλούνται **εικονικές διευθύνσεις** καθότι **δεν αντιστοιχούν σε διευθύνσεις της φυσικής μνήμης**
 - Αυτό γιατί μία σελίδα διαδικασίας μπορεί να βρεθεί σε οποιοδήποτε πλαίσιο σελίδας.
- Επομένως δεν μπορεί να υπάρξει μονοσήμαντη αντιστοιχία μεταξύ μιας διεύθυνσης που παράγει μία διαδικασία (εικονική διεύθυνση) και μιας φυσικής διεύθυνσης (physical address)
 - Πόσο μάλλον όταν η KM διαμοιράζεται μεταξύ πολλών διαδικασιών

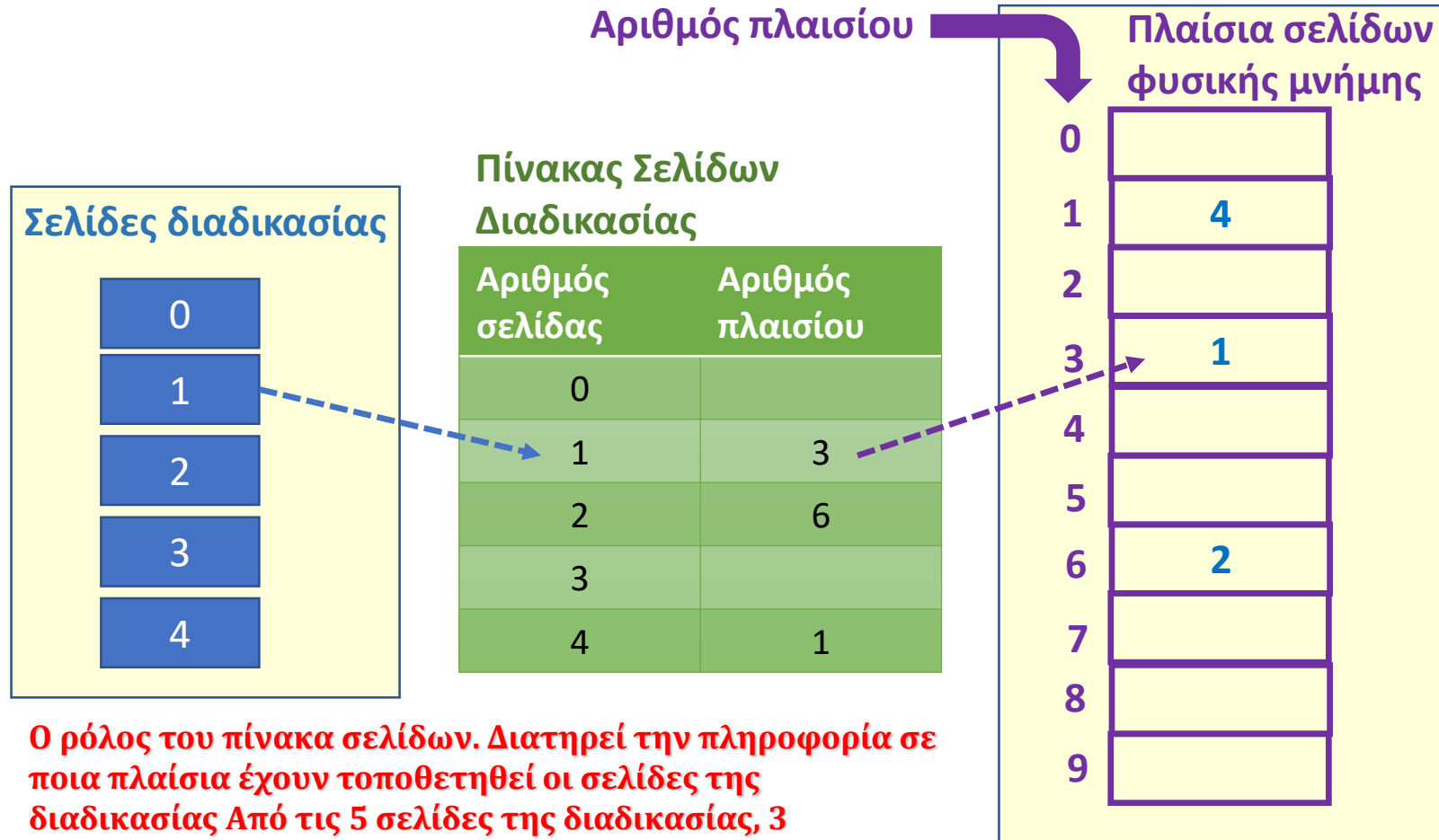
Σελιδοποίηση (Paging)

- Οι **εικονικές διευθύνσεις πρέπει να μετατραπούν σε φυσικές διευθύνσεις** προκειμένου να γίνει προσπάθεια σε ένα στοιχείο.
 - Οι εικονικές διευθύνσεις που παράγονται από την ΚΜΕ σε ένα σύστημα με σελιδοποίηση μπορούν να αναπαρασταθούν σε δισδιάστατη μορφή ως **(p, d)**, όπου p η σελίδα και d η μετατόπιση εντός της σελίδας.
 - Οι φυσικές διευθύνσεις στις οποίες μετατρέπονται οι εικονικές μπορούν να αναπαρασταθούν σε δισδιάστατη μορφή ως **(f, d)**, όπου f ο αριθμός πλαισίου μνήμης και d η μετατόπιση εντός του πλαισίου

Σελιδοποίηση (Paging)

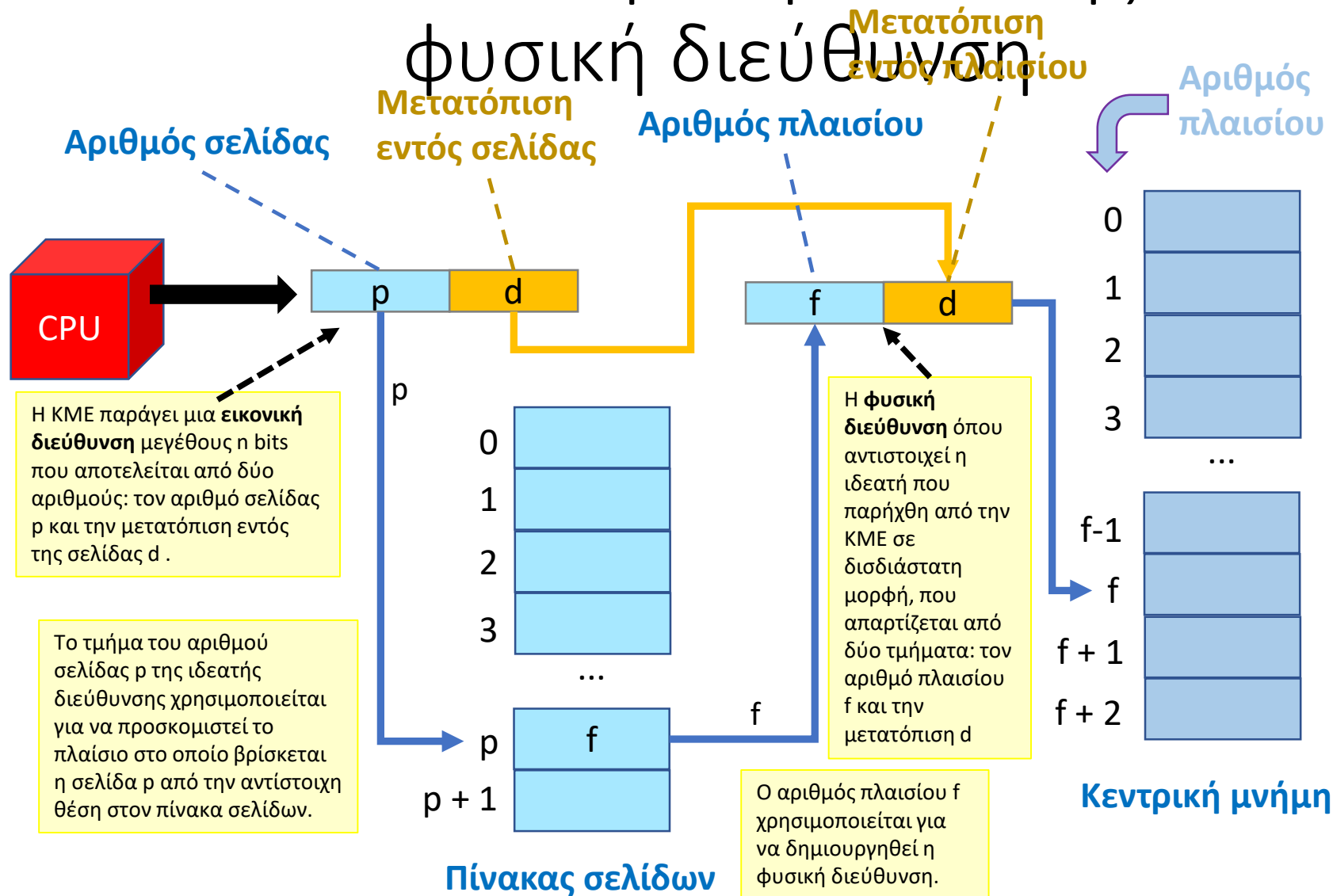


Ο ρόλος του Πίνακα σελίδων



Ο ρόλος του πίνακα σελίδων. Διατηρεί την πληροφορία σε ποια πλαίσια έχουν τοποθετηθεί οι σελίδες της διαδικασίας. Από τις 5 σελίδες της διαδικασίας, 3 βρίσκονται σε πλαίσια της κεντρικής μνήμης (1,2 και 4) ενώ 2 σελίδες δεν βρίσκονται στη μνήμη (0,3).

Σελιδοποίηση (Paging): Μετατροπή εικονικής σε φυσική διεύθυνση



Σελιδοποίηση (Paging)

- Συνηθίζεται επίσης, οι δισδιάστατες φυσικές διευθύνσεις που προκύπτουν από τις εικονικές και είναι της μορφής:

Αριθμός πλαισίου

Μετατόπιση

να εκφράζονται σε μονοδιάστατες, απόλυτες φυσικές διευθύνσεις μνήμης εφαρμόζοντας τον τύπο:

Φυσική διεύθυνση = <Αριθμός πλαισίου>* <Μέγεθος σελίδας> + <Μετατόπιση>

Παράδειγμα μετατροπής εικονικής σε φυσική διεύθυνση

Δίνεται ότι:

Μέγεθος σελίδας = 32Bytes

Ερώτηση:

Η εικονική διεύθυνση (3,15)
σε ποια φυσική διεύθυνση
αντιστοιχεί για το διπλανό
σχήμα;

Σελίδες
0
1
2
3

Διαδικασία

Σελίδα	Πλαίσιο
0	3
1	7
2	4
3	6

Πίνακας Σελίδων

Αριθμ
ός
πλαισί
ου

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

Κεντρική Μνήμη

Η εικονική διεύθυνση (3,15) σημαίνει ότι επιχειρείται να γίνει αναφορά σε ένα στοιχείο που βρίσκεται στη σελίδα 3 της διαδικασίας και εντός της σελίδας αυτής στη θέση (byte) 15 (μετατόπιση).

Από τον πίνακα σελίδων φαίνεται ότι η σελίδα 3 βρίσκεται στο πλαίσιο 6. Κατά συνέπεια, η φυσική διεύθυνση στην οποία αντιστοιχεί η εικονική (3, 15) είναι: **(6, 15)**. Η απόλυτη φυσική διεύθυνση όπου αντιστοιχεί η φυσική διεύθυνση (6, 15) μπορεί να υπολογιστεί από τον τύπο:

Φυσική διεύθυνση = Αριθμός Πλαισίου * Μέγεθος Σελίδας + Μετατόπιση = 6*32 + 15 = 207

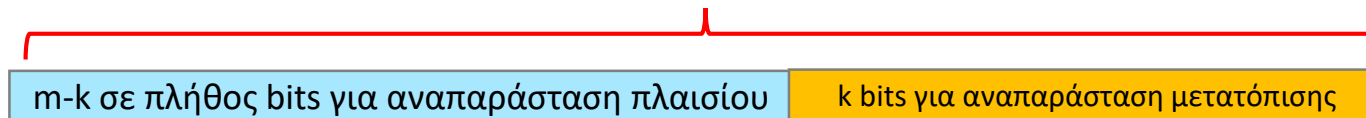
Σελιδοποίηση (Paging)

- Οι εικονικές και φυσικές διευθύνσεις συνηθίζεται να αναπαρίστανται σε bits, με συγκεκριμένο πλήθος bits εντός αυτές να αναπαριστούν την σελίδα ή πλαίσιο και τη μετατόπιση

Εικονική διεύθυνση- συνολικά (εύρος) n bits



Φυσική διεύθυνση- συνολικά (εύρος) m bits



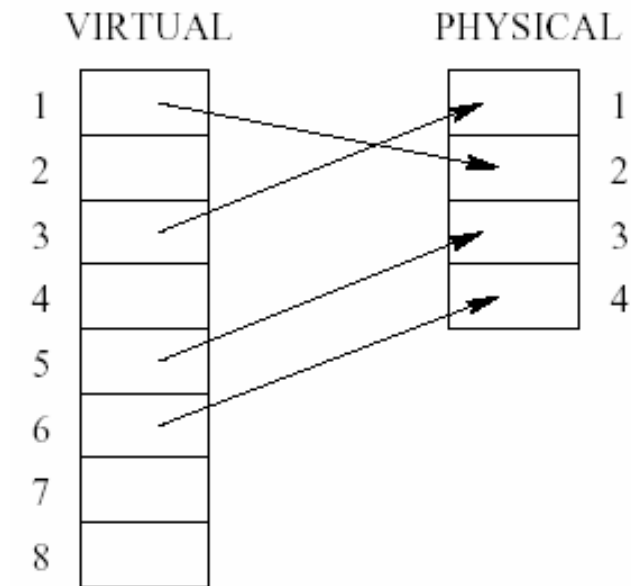
- Το **πλήθος των bits** που απαιτούνται για την **αναπαράσταση σελίδας/πλαισίου και της μετατόπισης** σε μία εικονική ή φυσική διεύθυνση, εξαρτάται από τα χαρακτηριστικά του συστήματος.

Σελιδοποίηση (Paging)

- Σε ένα σύστημα με **εικονικές διευθύνσεις εύρους n bits**, αυτό σημαίνει ότι μπορούν να διευθυσιοδοτηθούν (δλδ να παραχθούν διακριτές διευθύνσεις για) συνολικά 2^n bytes. Αυτό αποτελεί και το **μέγεθος του εικονικού χώρου διευθύνσεων**.
- Από τα n bits της εικονικής διεύθυνσης **εάν k bits από αυτά χρησιμοποιούνται για την μετατόπιση**, αυτό σημαίνει ότι μπορούν να δημιουργηθούν 2^k σε πλήθος διευθύνσεις εντός μιας σελίδας. Αυτό αποτελεί και το **μέγεθος της σελίδας**.
- Ο χώρος εικονικών διευθύνσεων, **σε πλήθος σελίδων**, θα είναι $2^n/2^k = 2^{n-k}$ σελίδες. Κατά συνέπεια σε εικονική διεύθυνση με n συνολικά bits, $n-k$ bits θα χρησιμοποιηθούν για την αναπαράσταση της σελίδας.
- Αν οι φυσικές διευθύνσεις έχουν **εύρος m bits**, τότε το μέγεθος της μνήμης (σε πλήθος διακριτών διευθύνσεων) θα είναι 2^m . Σε **πλήθος πλαισίων**, το μέγεθός της θα είναι $2^m/2^k = 2^{m-k}$ (στη σελιδοποίηση μέγεθος σελίδας = μέγεθος πλαισίου). Κατά συνέπεια k bits της φυσικής διεύθυνσης χρησιμοποιούνται για την μετατόπιση και τα υπόλοιπα $m-k$ για την αναπαράσταση του πλαισίου.

Σελιδοποίηση (Paging) Παράδειγμα μετατροπής εικονικής σε φυσική διεύθυνση

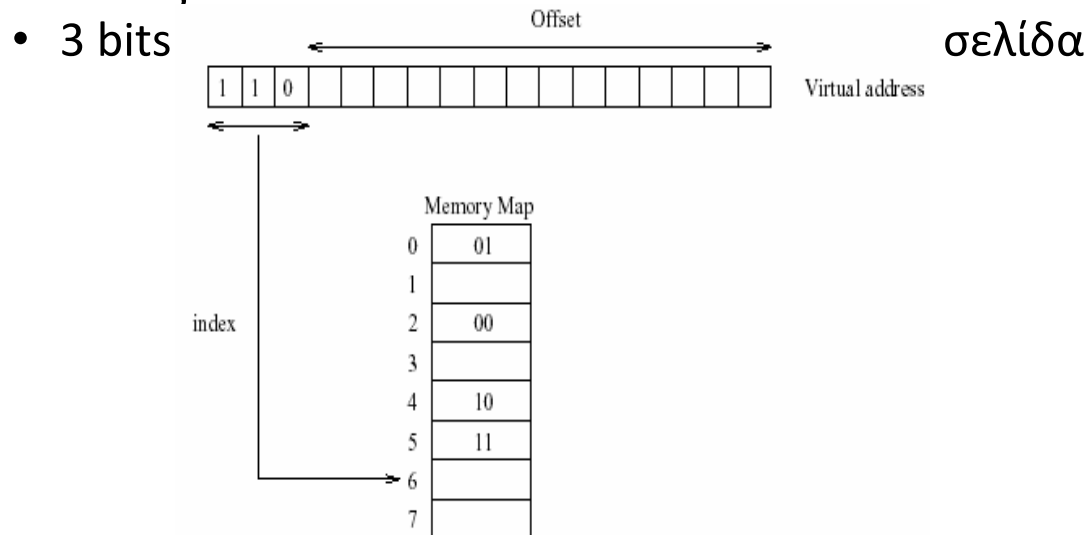
- Έστω υπολογιστικό σύστημα με
 - 16-bit διευθύνσεις
 - Χώρος εικονικών διευθύνσεων = $2^{16} = 64\text{Kbytes}$
 - 32 KBytes φυσική μνήμη
 - Μέγεθος σελίδας = 8Kbytes



Memory map

Μετατροπή εικονικής σε φυσική διεύθυνση: παράδειγμα

- Κάθε εικονική διεύθυνση έχει 16 bit
- Κάθε σελίδα είναι 8 Kbytes = 2^{13}
 - Απαιτούνται 13 bits για διευθυνσιοδότηση ενός byte μέσα σε κάθε σελίδα
- 64 Kbytes χώρος εικονικών διευθύνσεων / 8Kbytes ανά σελίδα
 - 8 σελίδες εικονικών διευθύνσεων



Μετατροπή εικονικής σε φυσική διεύθυνση

- Κατά συνέπεια
 - Τα 3 πρώτα bits μιας εικονικής διεύθυνσης μνήμης ορίζουν την εικονική σελίδα
 - Τα υπόλοιπα 13 bits της εικονικής διεύθυνσης ορίζουν το byte μέσα στην σελίδα

Μετατροπή εικονικής σε φυσική διεύθυνση: Άσκηση

- Σε ένα σύστημα διαχείρισης μνήμης με σελιδοποίηση, μια διεργασία διαθέτει τον παρακάτω πίνακα σελίδων (οι αριθμοί δίνονται στο δεκαεξαδικό σύστημα):

Αριθμός σελίδας	Αριθμός πλαισίου
0	A000
1	5000
2	7000
...	...
1E	2000
1F	C000

Το εύρος των λογικών διεύθυνσεων είναι 20 bits και το μέγεθος σελίδας είναι 4096 bytes (η διευθυνσιοδότηση γίνεται ανά byte). Έστω η λογική διεύθυνση 1EFFF. Σε ποια φυσική διεύθυνση αντιστοιχεί;

Μετατροπή εικονικής σε φυσική διεύθυνση: Παράδειγμα

- Με βάση τα δεδομένα της εκφώνησης, από τη στιγμή που το μέγεθος σελίδας είναι $4096=2^{12}$ bytes, μπορούμε να συμπεράνουμε ότι η **‘μετατόπιση’** αποτελείται από **12 bits**. Κατά συνέπεια επίσης ο **‘αριθμός σελίδας’** θα αποτελείται από $20 - 12 = 8$ bits.
- Μετατρέπουμε τη δοθείσα διεύθυνση στο δυαδικό και διαχωρίζουμε τον αριθμό σελίδας και τη μετατόπιση σύμφωνα με τα παραπάνω:
- **1EFFC** = 0001 1110 1111 1111 1100
- Αριθμός σελίδας: 0001 1110 = **1E**
- Μετατόπιση: 1111 1111 1100 = **FFC**
- Με βάση τον **Πίνακα Σελίδων** η λογική σελίδα **1E** αντιστοιχεί στη φυσική σελίδα (πλαίσιο): **2000**
- Άρα η τελική φυσική διεύθυνση είναι η:

0010 0000 0000 0000 1111 1111 1100 = **2000FFC (hex)**

Τμηματοποίηση (Segmentation)

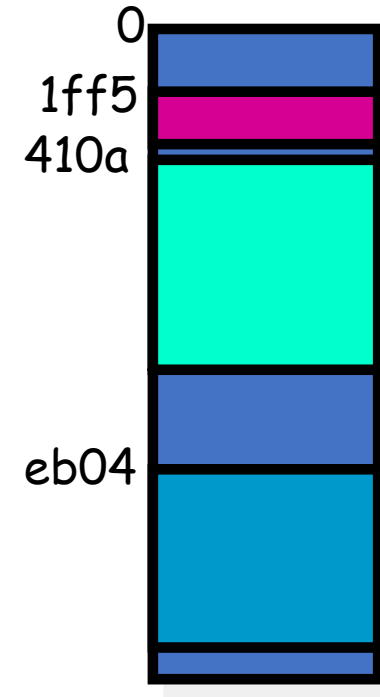
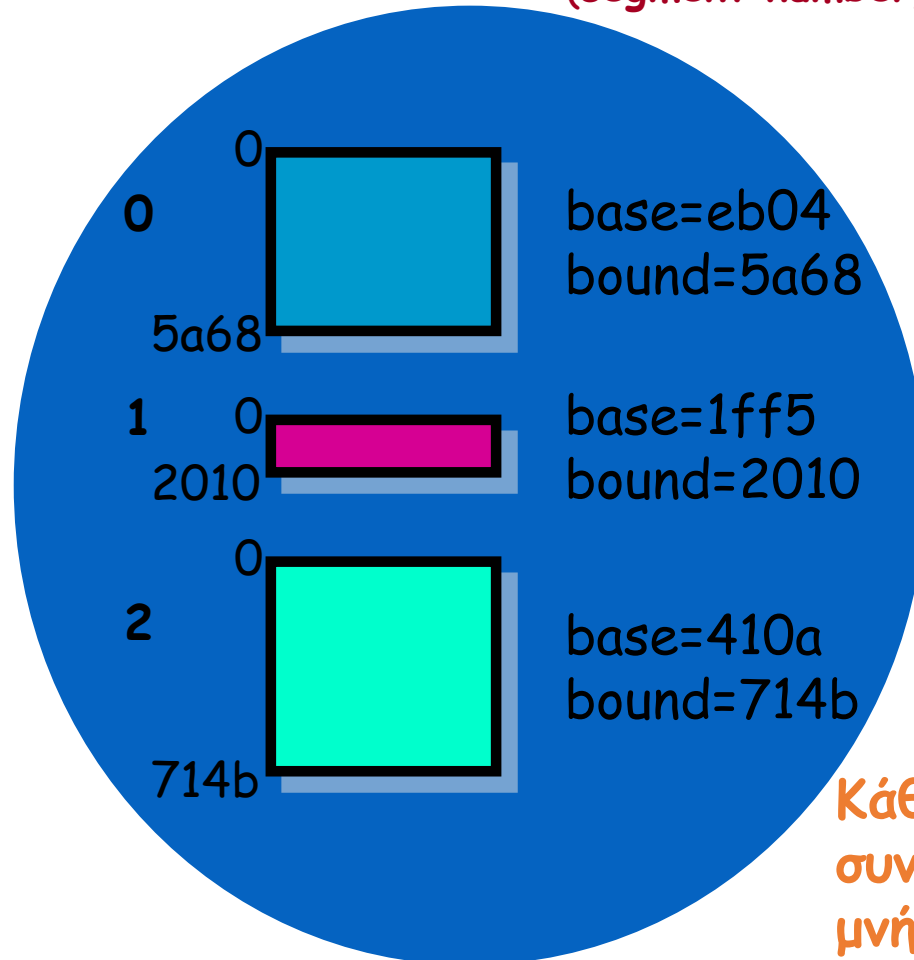
- Ένα τμήμα (segment) είναι ένα μεταβλητού μεγέθους σύνολο συνεχόμενων διευθύνσεων μνήμης στον (ιδεατό) χώρο διευθύνσεων μιας διεργασίας που οργανώνεται και διαχειρίζεται από το ΛΣ ως μια ενιαία μονάδα.
- **Τμηματοποίηση** είναι ο τρόπος οργάνωσης της ιδεατής μνήμης σε τμήματα.
- Μια διεύθυνση αποτελείται από δύο μέρη – έναν αριθμό τμήματος και μια μετατόπιση (offset).

Τμηματοποίηση (Segmentation)

- Τα τμήματα δεν είναι ίσα και η τμηματοποίηση είναι παρόμοια με τη δυναμική υποδιαίρεση
 - Μειώνεται ο εσωτερικός κατακερματισμός
- Τα τμήματα μπορούν να έχουν δυναμικό (μεταβλητό) μέγεθος ώστε να διευκολύνεται η διαχείριση δυναμικών δομών δεδομένων
- Η τμηματοποίηση:
 - Επιτρέπει στα προγράμματα να τροποποιούνται και να μεταφράζονται εκ νέου ανεξάρτητα
 - Είναι κατάλληλη για διαμοίραση και προστασία δεδομένων

Τμηματοποίηση (Segmentation)

Μια εικονική διεύθυνση είναι ένας αριθμός τμήματος (segment number) και μια μετατόπιση offset.



Κάθε τμήμα τοποθετείται σε μια συνεχόμενη περιοχή της μνήμης.

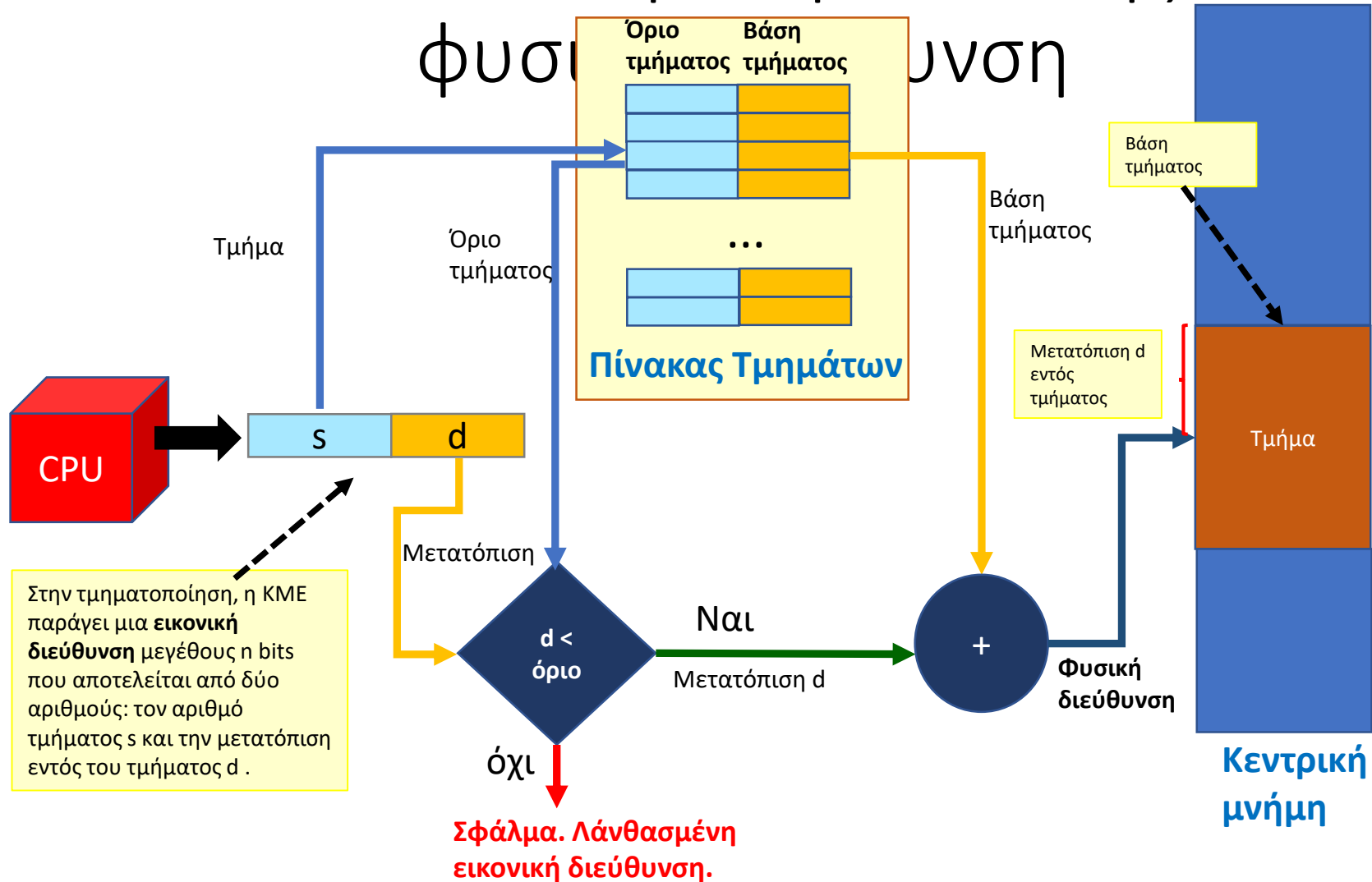
Τμηματοποίηση (Segmentation)

- Πίνακας τμημάτων
 - Διατηρεί τις πληροφορίες για τα τμήματα εντός της κεντρικής μνήμης.
 - Για κάθε τμήμα, ο πίνακας τμημάτων διατηρεί:
 - **το μέγεθος του τμήματος** – καλείται και όριο του τμήματος (limit)
 - **τη βάση του τμήματος (Base)** – την πρώτη διεύθυνση από όπου ξεκινά το συγκεκριμένο τμήμα εντός της κεντρικής μνήμης
 - **Επιπλέον πληροφορίες για την διαχείριση** όπως αν το τμήμα έχει φορτωθεί στη μνήμη, αν έχει τροποποιηθεί κλπ

Όριο τμήματος	Βάση τμήματος	Έχει φορτωθεί;	Έχει τροποποιηθεί;
150	200	1	0
100	400	0	0
80	600	1	1

- Ο πίνακας διατηρεί τις πληροφορίες για τα τμήματα εντός της κεντρικής μνήμης ή εικονικών διευθύνσεων
- Κάθε διαδικασία έχει τον δικό της πίνακα τμημάτων
- Η τελευταία διεύθυνση που ανήκει σε ένα τμήμα είναι η:
 - Βάση Τμήματος + Μέγεθος Τμήματος

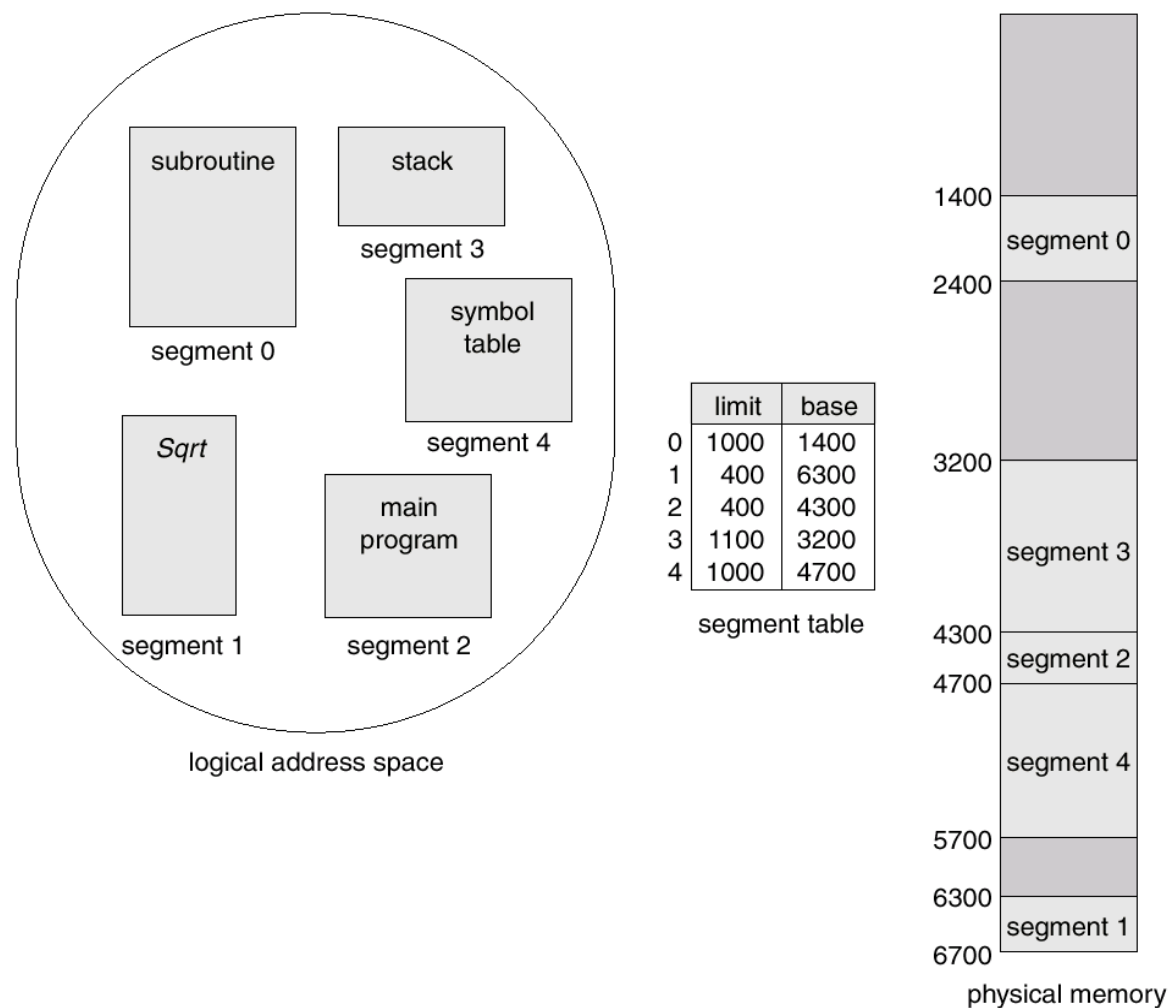
Τμηματοποίηση (Segmentation): Μετατροπή εικονικής σε φυσική διεύθυνση



Τμηματοποίηση (Segmentation)

- Η τμηματοποίηση είναι φανερή στον προγραμματιστή σε αντίθεση με τη σελιδοποίηση
 - Παρέχεται ως διευκόλυνση για την οργάνωση προγραμμάτων και δεδομένων
- Ο προγραμματιστής βλέπει το πρόγραμμα σαν συλλογή από τμήματα, π.χ. main program, function, objects, global variables, stack, ...
- Δεν υπάρχει μια απλή συσχέτιση μεταξύ των λογικών και των φυσικών διευθύνσεων

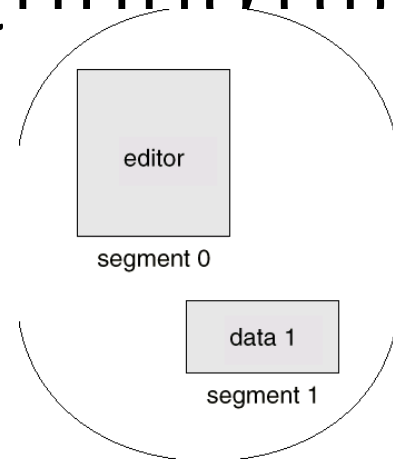
Παράδειγμα συστήματος με τμήματα



Πλεονεκτήματα της τμηματοποίησης

- Η εικόνα της μνήμης είναι η εικόνα που έχει ο προγραμματιστής (ή ο έμπειρος χρήστης)
- Τα τμήματα προστατεύονται μεταξύ τους
 - Κάθε τμήμα περιέχει ένα τύπο πληροφορίας (π.χ. instructions, stack, ...)
- Η διαμοίραση τμημάτων είναι λογική και εύκολη
 - Αν όλες οι εντολές είναι σε ένα τμήμα και όλα τα δεδομένα σε άλλο, το τμήμα εντολών μπορεί να διαμοιραστεί ελεύθερα σε διαφορετικές διεργασίες (κάθε μια με τα δικά της δεδομένα)

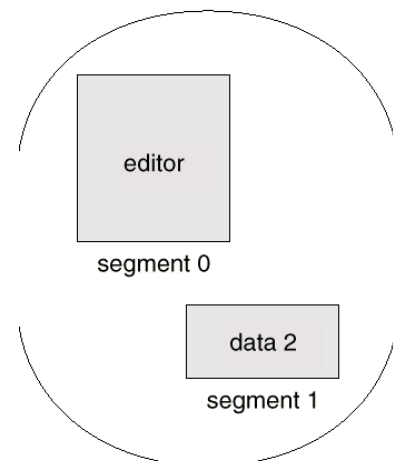
Διαμοίραση τιμηιότητων



logical memory
process P_1

	limit	base
0	25286	43062
1	4425	68348

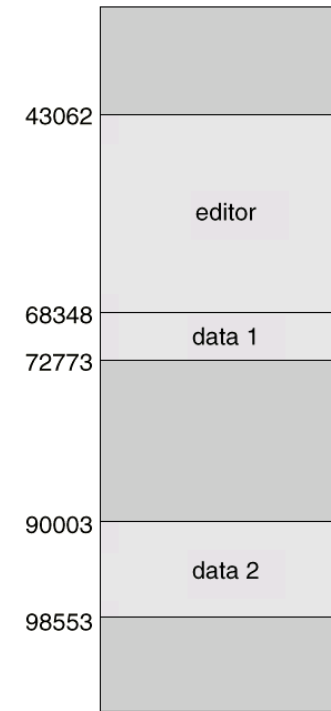
segment table
process P_1



logical memory
process P_2

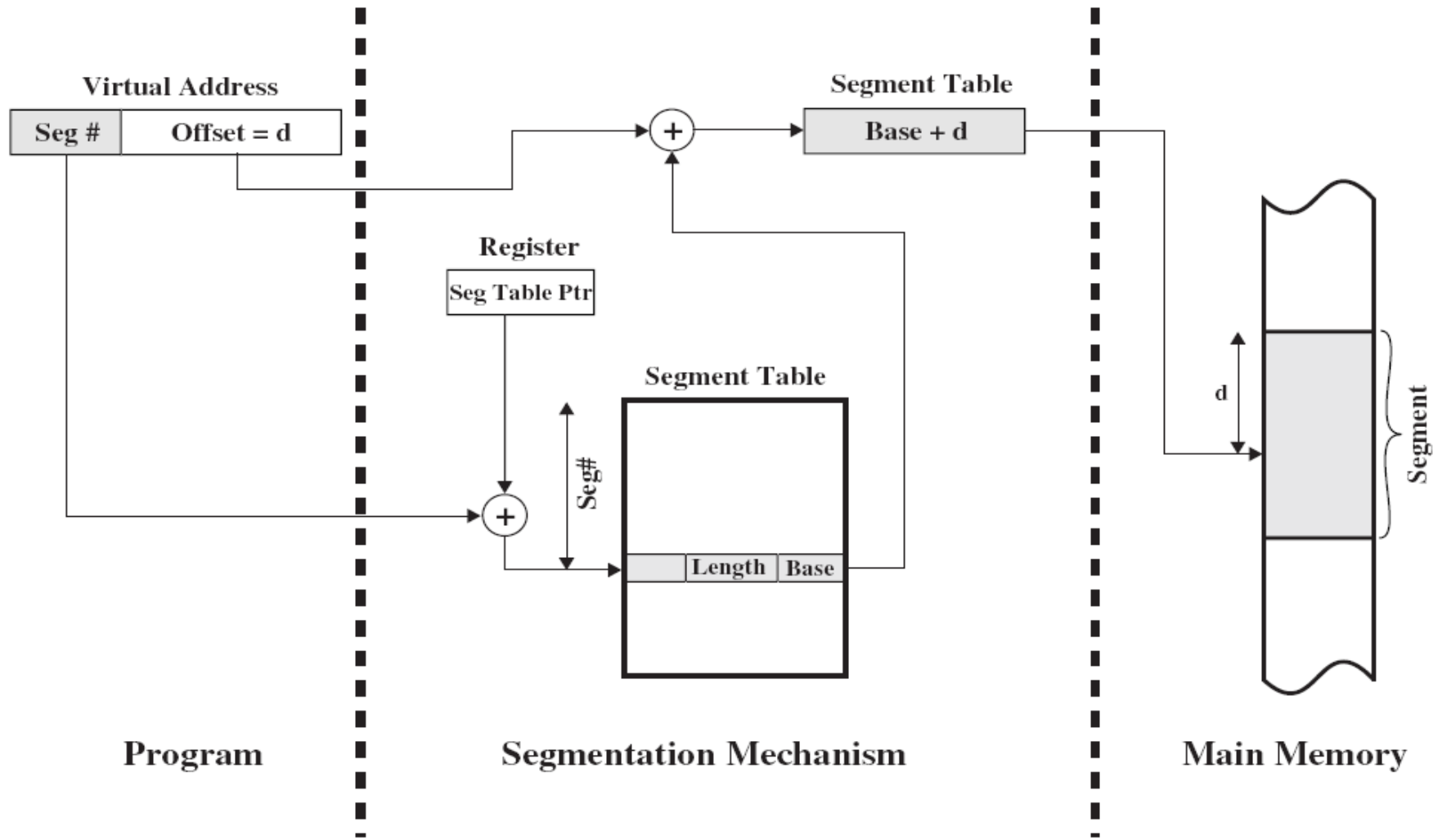
	limit	base
0	25286	43062
1	8850	90003

segment table
process P_2

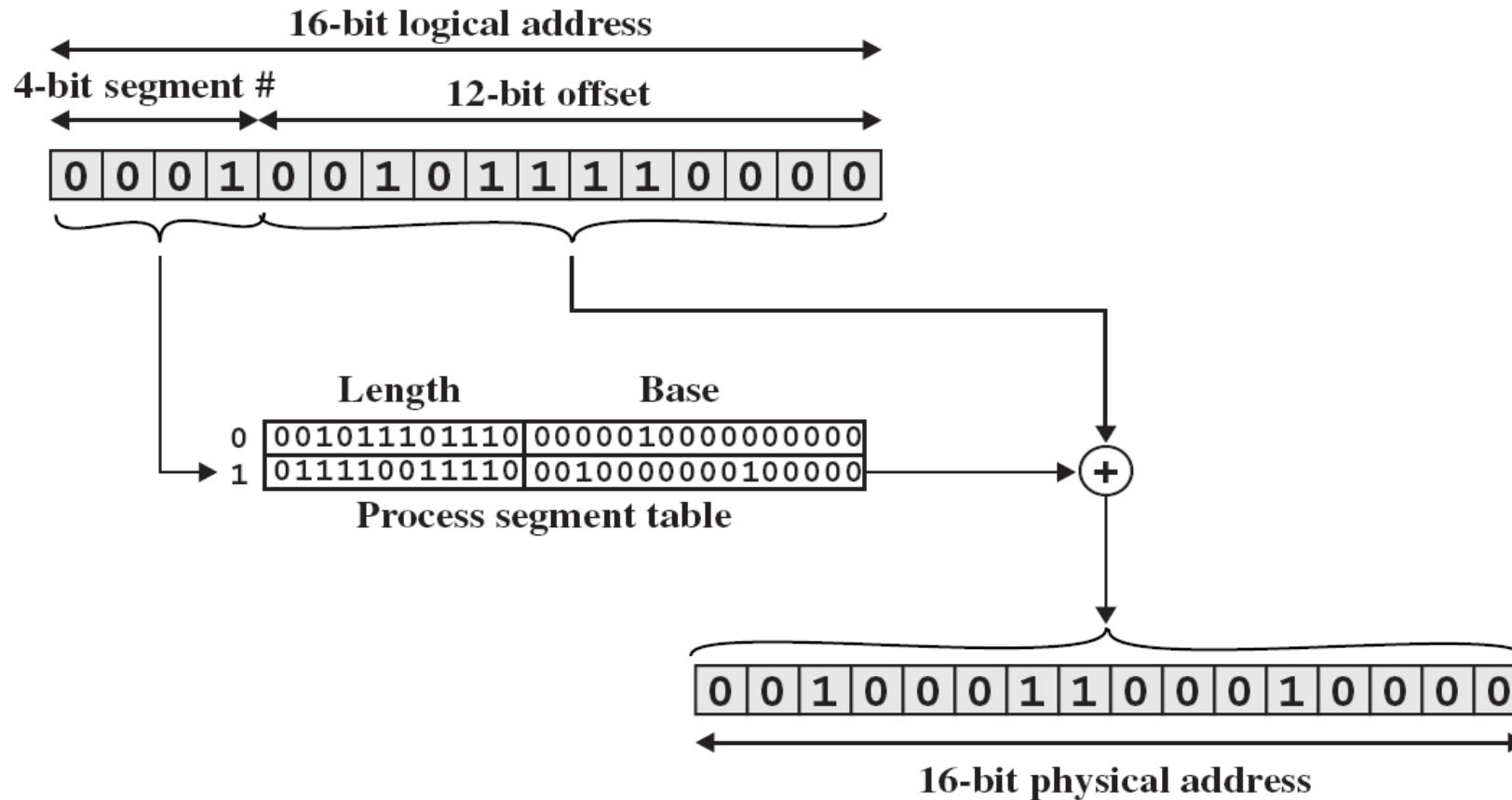


physical memory

Μετατροπή διεύθυνσης σε σύστημα με τμηματοποίηση



Τμηματοποίηση : μετατροπή λογικών σε φυσικές διευθύνσεις



Ανάλυση του παραδείγματος

- Λογική διεύθυνση : 0001 0010 1111 0000
 - Αριθμός τμήματος : 4 πρώτα bits και είναι ο αριθμός 0001
- Μετατόπιση τμήματος τα υπόλοιπα 12 bits
 - Μέγιστο μέγεθος τμήματος $2^{12} = 4096$ bytes

2048	1024	512	256	128	64	32	16	8	4	2	1
0	0	1	0	1	1	1	1	0	0	0	0

**12 Bits μετατόπισης της λογικής διεύθυνσης
0001 0010 1111 0000**

Μετάφραση σε φυσική διεύθυνση

- Αριθμός τμήματος 0001
- Ο αριθμός αυτός χρησιμοποιείται στον πίνακα τμημάτων της διαδικασίας για να βρεθεί η φυσική διεύθυνση της αρχής του τμήματος
- Συγκρίνεται η μετατόπιση με το μήκος του τμήματος
 - Αν Μετατόπιση > Μήκος τμήματος
 - ΜΗ ΕΓΚΥΡΗ ΔΙΕΥΘΥΝΣΗ
- Διαφορετικά: Φυσική διεύθυνση = Φυσική διεύθυνση αρχής τμήματος+Μετατόπιση

Μετάφραση σε φυσική διεύθυνση

2048	1024	512	256	128	64	32	16	8	4	2	1
0	1	1	1	1	0	0	1	1	1	1	0

- Από τον πίνακα τμημάτων προκύπτει ότι το τμήμα 1 έχει μήκος : $1024 + 512 + 256 + 128 + 16 + 4 + 2 = 1950$ (βλ. διαφάνεια 121)
- Η μετατόπιση στην περίπτωση μας είναι : $512 + 128 + 64 + 32 + 16 = 752 < 1950$
 - Άρα η διεύθυνση είναι έγκυρη
- Για να βρεθεί η φυσική διεύθυνση θα προσθέσω τη μετατόπιση στη βάση του τμήματος 1

base	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
offset	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0
address	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0

Σελιδοποιημένη τμηματοποίηση (Paged segmentation)

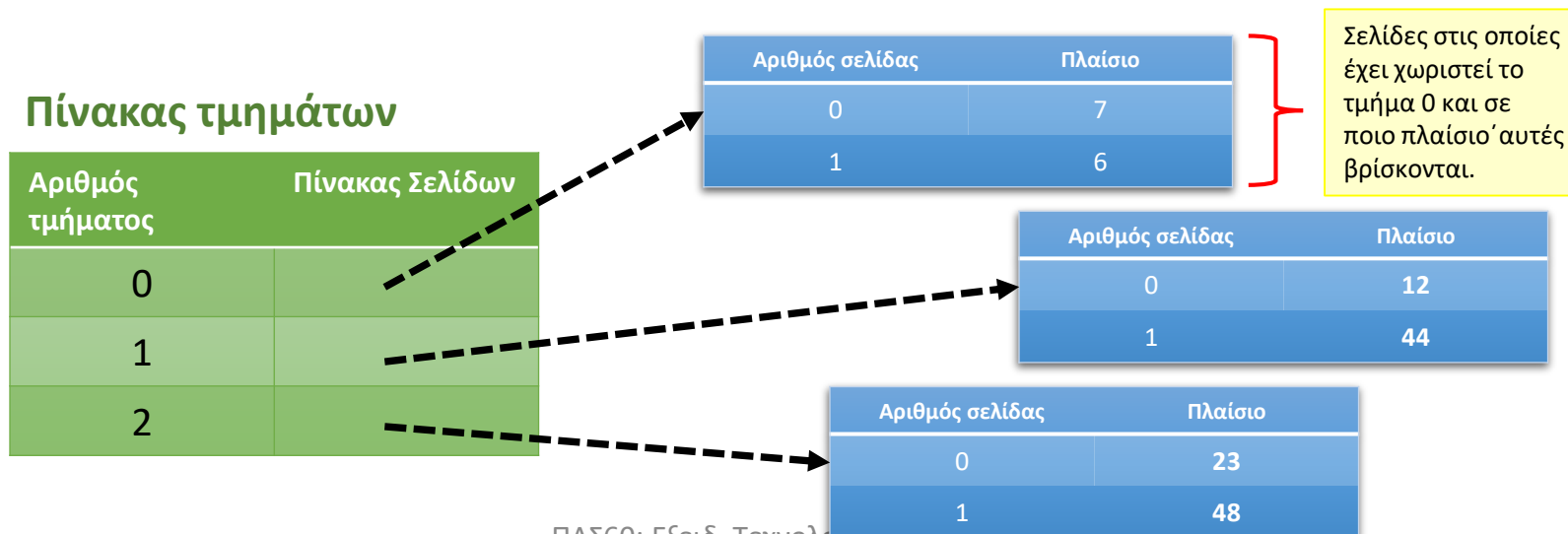
- Αριθμός τμήματος 0001
- Ο αριθμός αυτός χρησιμοποιείται στον πίνακα τμημάτων της διαδικασίας για να βρεθεί η φυσική διεύθυνση της αρχής του τμήματος
- Συγκρίνεται η μετατόπιση με το μήκος του τμήματος
 - Αν Μετατόπιση > Μήκος τμήματος
 - ΜΗ ΕΓΚΥΡΗ ΔΙΕΥΘΥΝΣΗ
- Διαφορετικά: Φυσική διεύθυνση = Φυσική διεύθυνση αρχής τμήματος+Μετατόπιση

Σελιδοποιημένη Τμηματοποίηση (Segmented Paging)

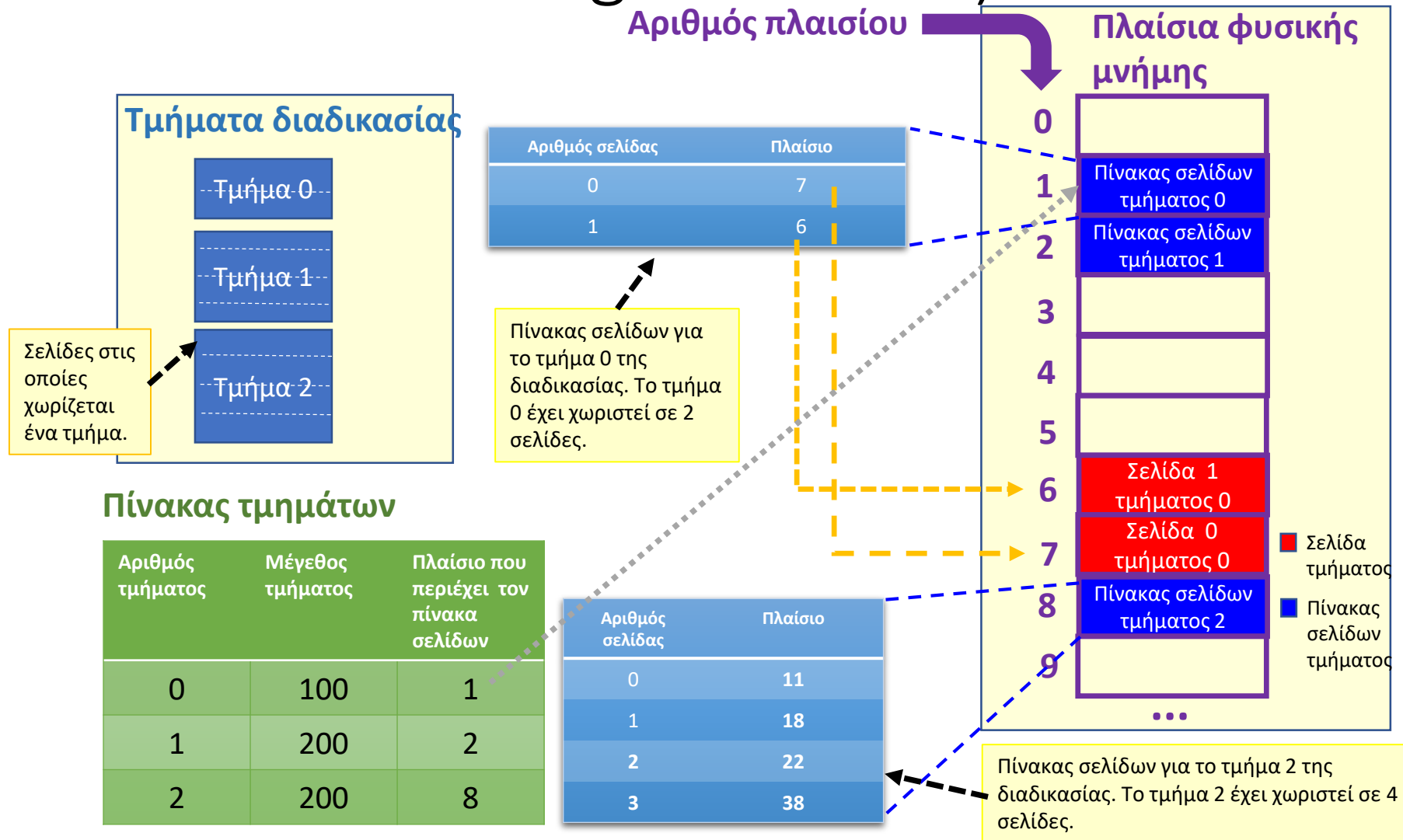
- Σελιδοποιημένη τμηματοποίηση
 - Αποτελεί τεχνική που κάνει χρήση συνδυασμού τμηματοποίησης και σελιδοποίησης για τη διαχείριση της μνήμης. Καλείται επίσης και τμηματοποίηση με σελίδες (Segmentation with paging)
 - Ο **εικονικός χώρος** της διαδικασίας **χωρίζεται σε τμήματα** - όπως στην τμηματοποίηση.
 - **Κάθε τμήμα** του εικονικού χώρου διευθύνσεων **χωρίζεται σε σελίδες** σταθερού μεγέθους - όπως στην σελιδοποίηση.
 - Από εκεί και το όνομα: σελιδοποιημένη τμηματοποίηση => **τμήματα που έχουν σελιδοποιηθεί.**
 - **Κεντρική μνήμη χωρίζεται σε πλαίσια**, ίσου μεγέθους με τις σελίδες στις οποίες χωρίζονται τα τμήματα.
 - Κάθε διαδικασία έχει έναν (1) **πίνακα τμημάτων** για την διατήρηση των τμημάτων.
 - **Κάθε τμήμα** του εικονικού χώρου διευθύνσεων έχει τον **δικό της πίνακα σελίδων** όπου διατηρούνται οι σελίδες στις οποίες έχει χωριστεί το συγκεκριμένο τμήμα.

Σελιδοποιημένη Τμηματοποίηση (Segmented Paging)

- Ο πίνακας σελίδων ενός τμήματος διατηρεί την πληροφορία, σε ποιο πλαίσιο βρίσκεται κάθε σελίδα του τμήματος
- Ο πίνακας σελίδων διατηρείται και αυτός σε πλαίσια της κεντρικής μνήμης
 - Ο **πίνακας τμημάτων** διατηρεί γραμμές που αναφέρουν σε ποιο σημείο της κεντρικής μνήμης αποθηκεύθηκε ο πίνακας σελίδων για το συγκεκριμένο τμήμα όπως και το μήκος του τμήματος. Ο κύριος ρόλος του πίνακα τμημάτων είναι να παραπέμψει στον κατάλληλο πίνακα σελίδων κάθε τμήματος.
 - Ο πίνακας τμημάτων μπορεί να διατηρεί είτε τον **αριθμό πλαισίου** ή τη **διεύθυνση βάσης** από όπου ξεκινά ο πίνακας σελίδων του συγκεκριμένου τμήματος εντός της μνήμης.

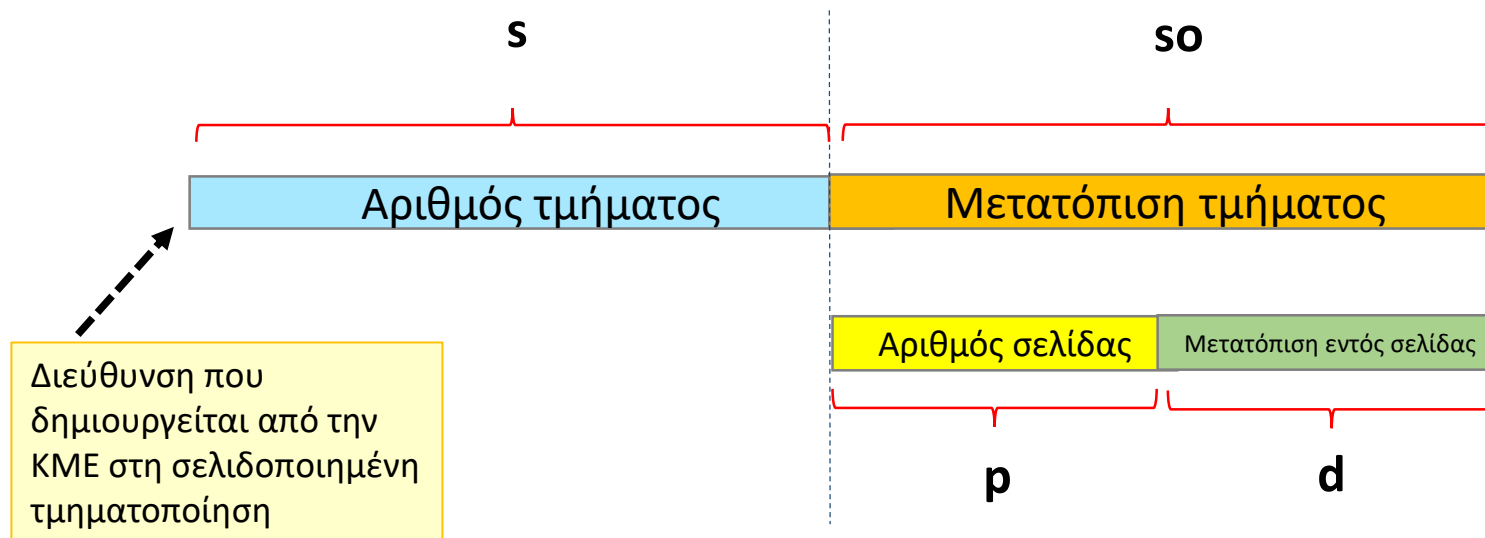


Σελιδοποίηση με τη τμηματοποίηση (Paged Segmentation)



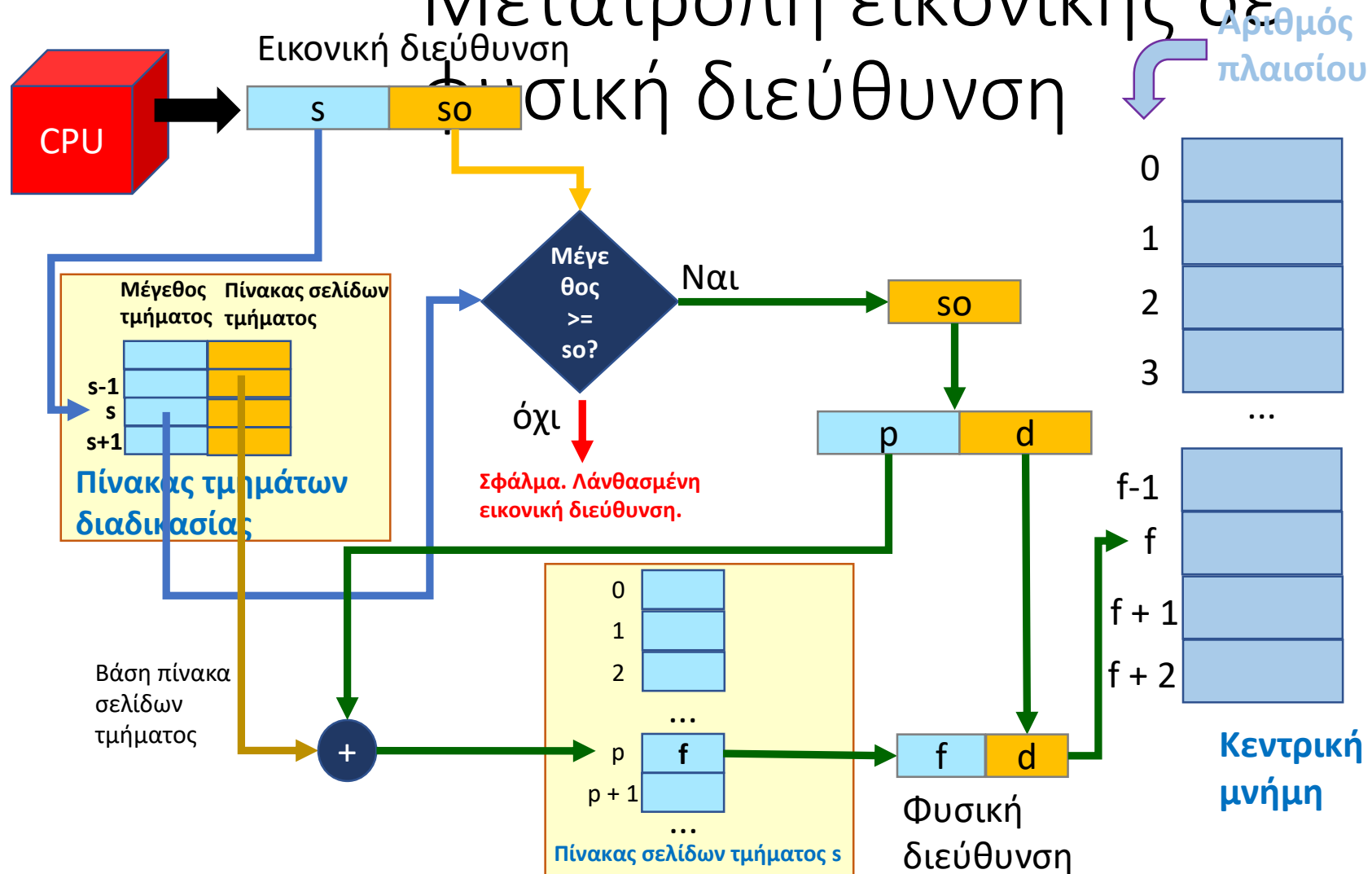
Σελιδοποιημένη Τμηματοποίηση (Segmented Paging): Διευθύνσεις

- Διευθύνσεις
 - Στη σελιδοποιημένη τμηματοποίηση η ΚΜΕ παράγει μία **δισδιάστατη εικονική διεύθυνση** της μορφής (s, so) όπου s ο αριθμός τμήματος και so η μετατόπιση (εντός του) τμήματος.
 - Η μετατόπιση τμήματος, so , με τη σειρά της **χωρίζεται σε δύο μέρη** (p, d) : τη σελίδα p και τη μετατόπιση εντός σελίδας d



τμηματοποίηση (Paged segmentation):

Μετατροπή εικονικής σε φυσική διεύθυνση



Σελιδοποιημένη τμηματοποίηση (Paged Segmentation)

- Χρησιμοποιείται σε σύγχρονες αρχιτεκτονικές
 - Π.χ. Intel Pentium
- Προτερήματα
 - Αποφεύγει εξωτερική κλασματοποίηση
 - Πίνακες σελίδων μπορούν να μοιραστούν μεταξύ διαδικασιών
- Μειονεκτήματα
 - Εμφανίζει εσωτερική κλασματοποίηση
 - 2 με 3 τάξεις μεγέθους πιο αργή διαδικασία καθότι απαιτείται περισσότερες πράξεις για την μετατροπή μιας εικονικής διεύθυνσης σε φυσική.

Πλάνο παρουσίασης

- Διαχείριση μνήμης
- Ιδεατή μνήμη

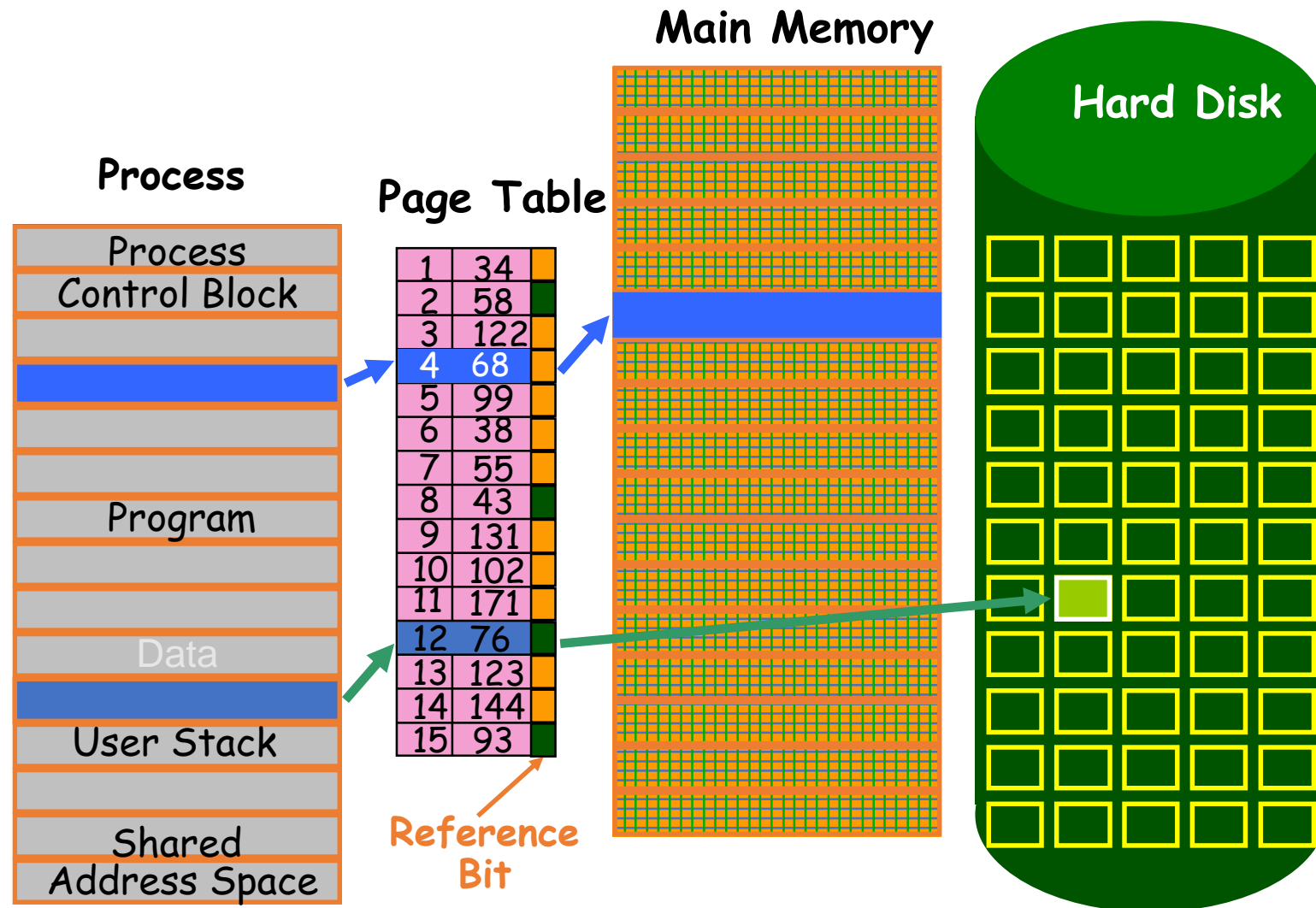
Η έννοια της ιδεατής μνήμης

- Σε ένα υπολογιστικό σύστημα με σελιδοποίηση
 - Κάθε διαδικασία θεωρεί πως έχει διαθέσιμο όλο το χώρο εικονικών διευθύνσεων
 - Κάποιες διευθύνσεις μνήμης αυτού του χώρου έχουν αντιστοίχιση με διευθύνσεις μνήμης στην φυσική μνήμη
 - Πολλαπλές διαδικασίες μπορούν να εκτελούνται ή/και να είναι έτοιμες προς εκτέλεση
- Τι συμβαίνει αν το σύνολο των πλαισίων που απαιτούν οι διαδικασίες είναι μεγαλύτερο από το σύνολο πλαισίων της φυσικής μνήμης;

Ιδεατή μνήμη (1)

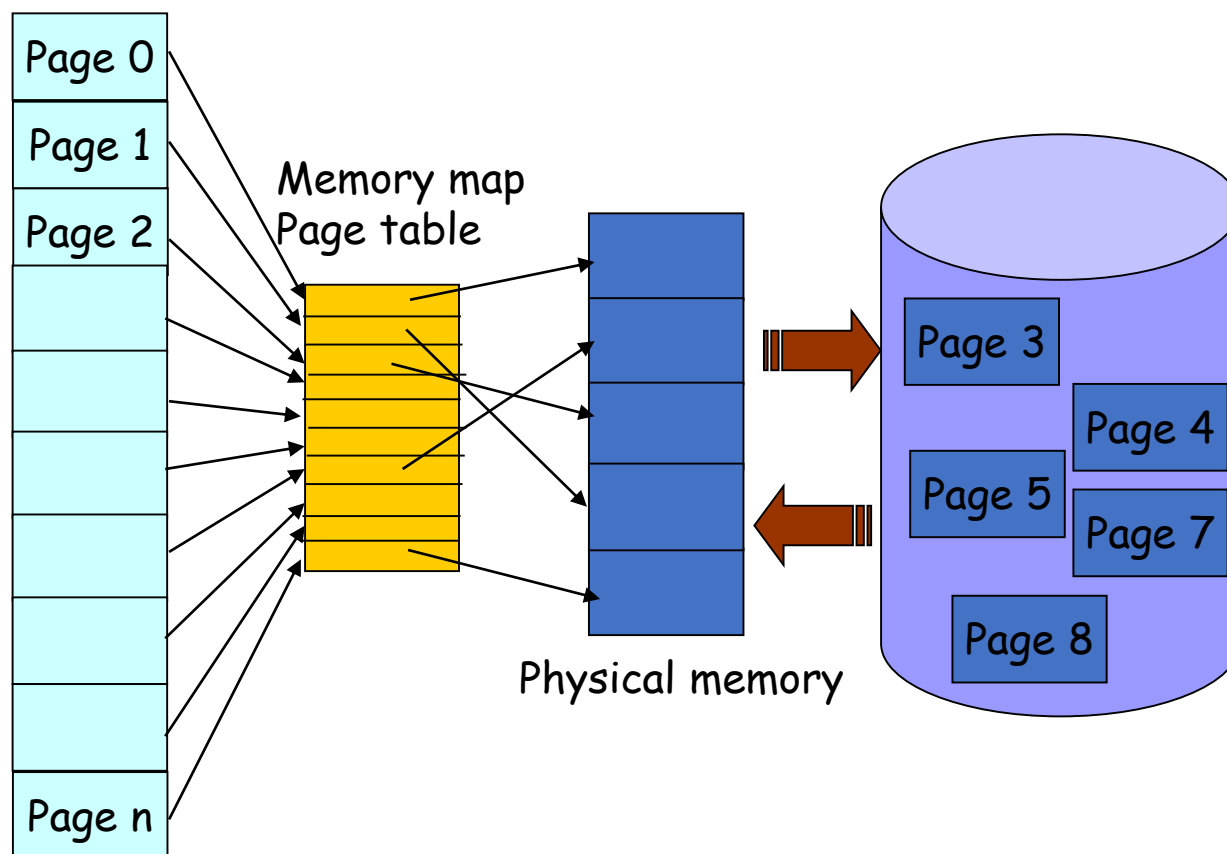
- Κάθε χρονική στιγμή ένα υποσύνολο των διαδικασιών εκτελείται πραγματικά
- Για κάθε διαδικασία που εκτελείται μόνο ένα υποσύνολο των πλαισίων σελίδας χρησιμοποιείται σε δεδομένο χρονικό διάστημα
- Τα υπόλοιπα πλαίσια σελίδων δεν χρειάζεται να είναι στην μνήμη (αν δεν χωράνε)
 - Αποθήκευση στον σκληρό δίσκο μέχρι να χρειαστούν

Ιδεατή μνήμη (2)



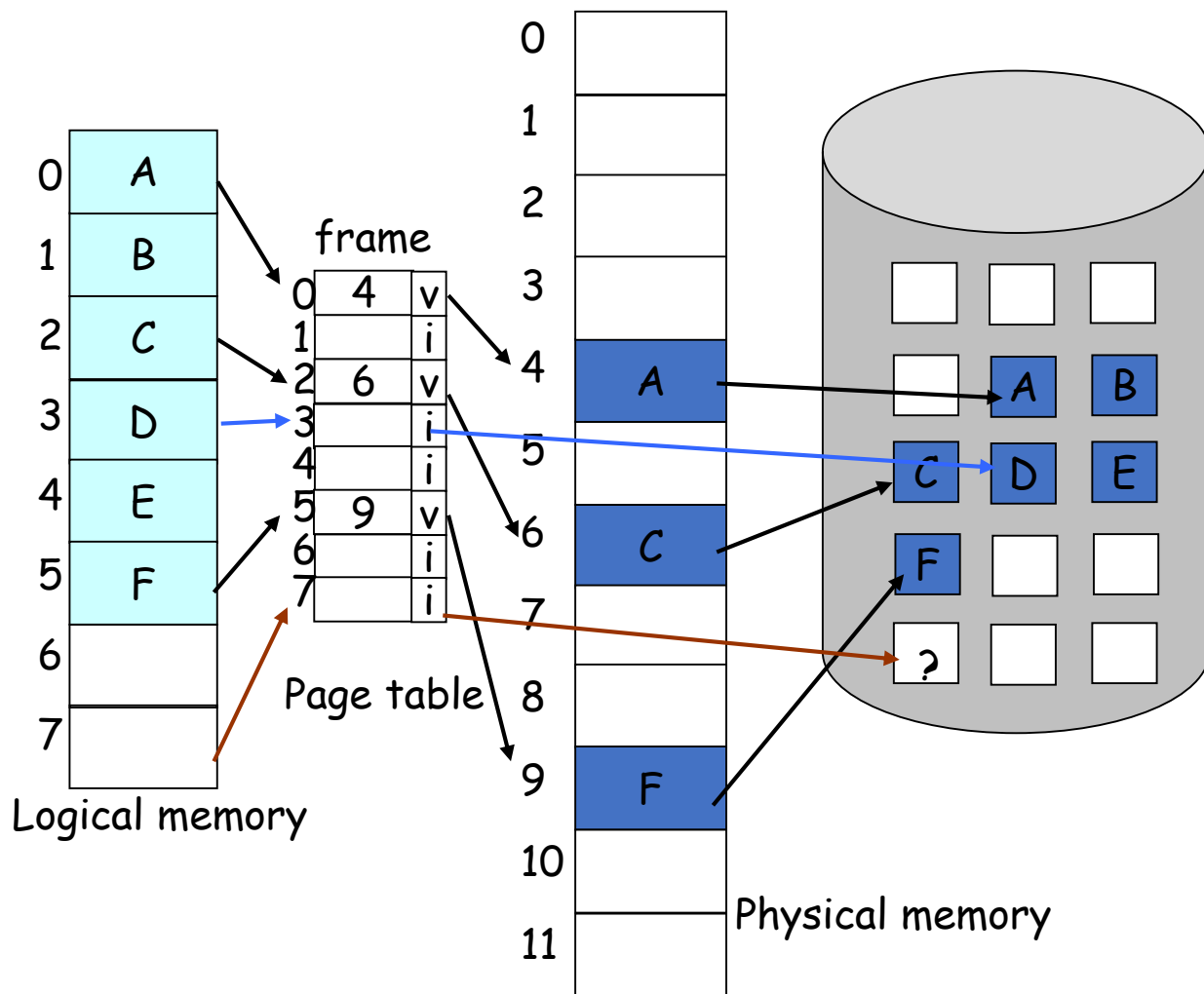
Ιδεατή μνήμη (3)

Virtual memory



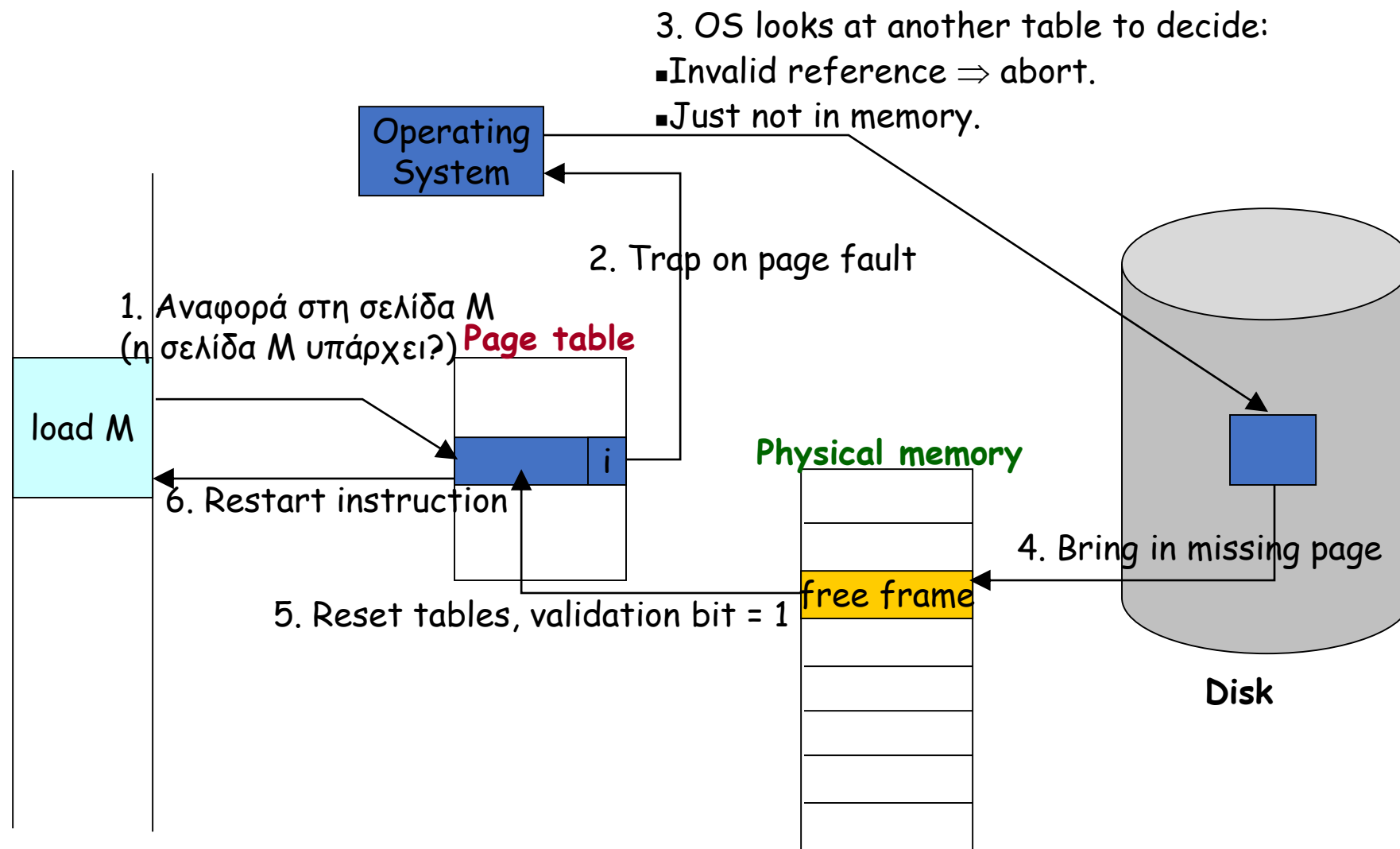
- Αναφορά σελίδας
- Έλεγχος χάρτη μνήμης
- Αν υπάρχει αντίστοιχο σημείο εισόδου στον πίνακα σελίδων για πλαίσιο φυσικής μνήμης τότε γίνεται αναφορά σε αυτό
- Διαφορετικά, η σελίδα δεν υπάρχει στη φυσική μνήμη και πρέπει να μεταφερθεί από τον σκληρό δίσκο
- Αν η φυσική μνήμη είναι πλήρης,
 - Εύρεση σελίδας προς αντικατάσταση
 - Εναλλαγή στον δίσκο

Διαχείριση πίνακα σελίδων ιδεατής μνήμης



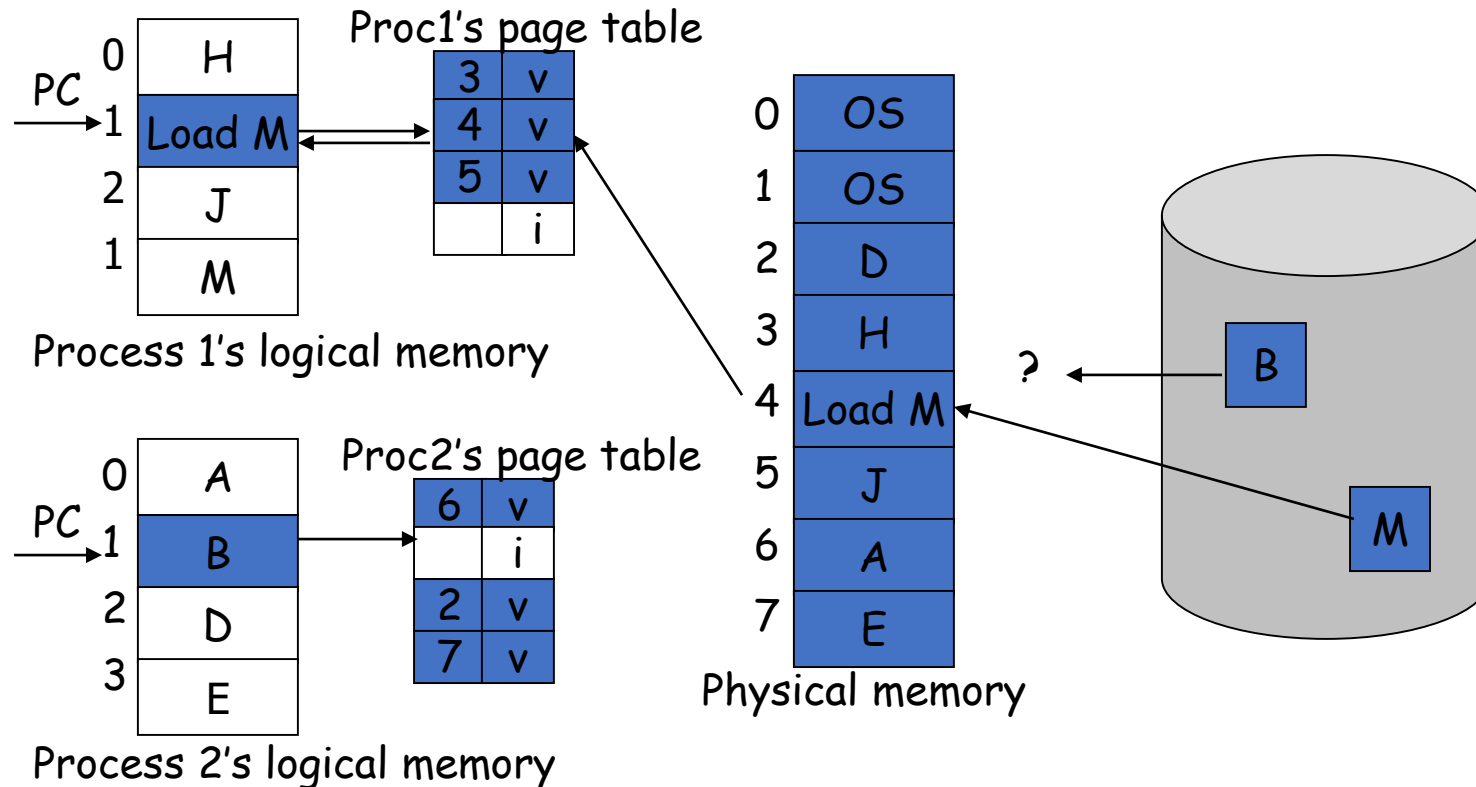
- Αν το bit πλαισίου είναι έγκυρο (valid), το πλαίσιο βρίσκεται στη μνήμη
- Διαφορετικά, συμβαίνει σφάλμα σελίδας
 - Η αντίστοιχη σελίδα φορτώνεται από τον δίσκο
 - Εκχωρείται μια νέα περιοχή μνήμης.

Σφάλμα σελίδας

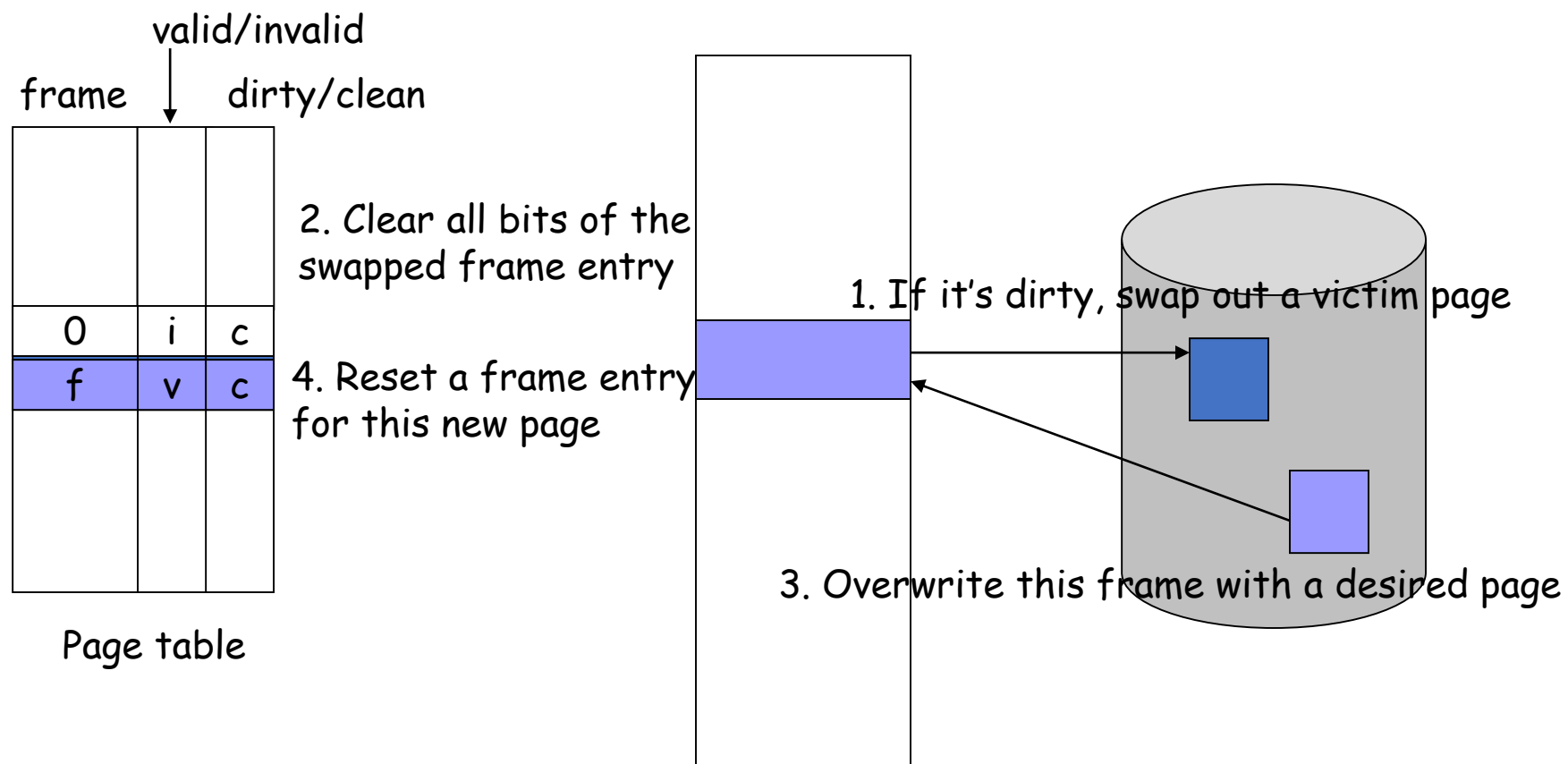


Αντικατάσταση σελίδας

- 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



Μηχανισμός αντικατάστασης σελίδας



Πολιτική αντικατάστασης

- 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

- Επτά (7) σελίδες
- Τέσσερα (4) πλαίσια

Ακολουθία 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	2	5	5	5	5	5	5
	3	3	3	3	3	3	6	6	6	6	6	6	1	1	3	3
		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
F	F	F	F		F		F	F	F		F		F	F		F

Σφάλματα σελίδων: 12

Παράδειγμα 2

- Επτά (7) σελίδες
- Πέντε (5) πλαίσια

Ακολουθία αναφοράς

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

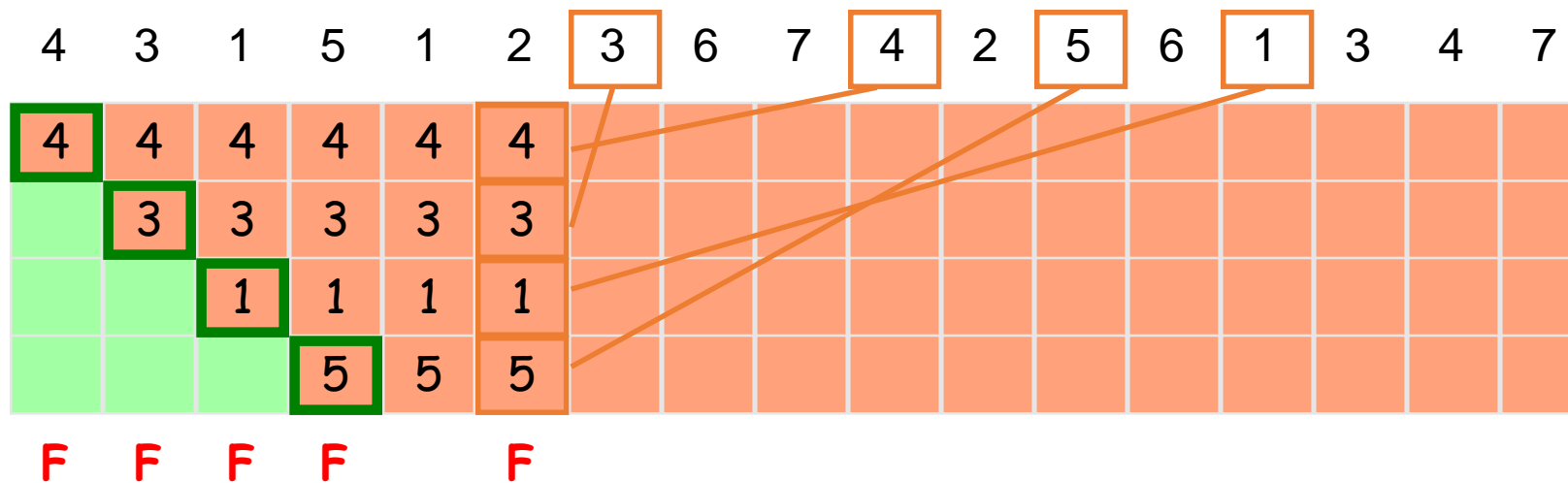
Σφάλματα σελίδας: 10

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 δεν χρησιμοποιείται σε πραγματικά συστήματα
- Χρησιμοποιείται ως benchmark για την αποτίμηση της απόδοσης άλλων αλγορίθμων αντικατάστασης

Παράδειγμα

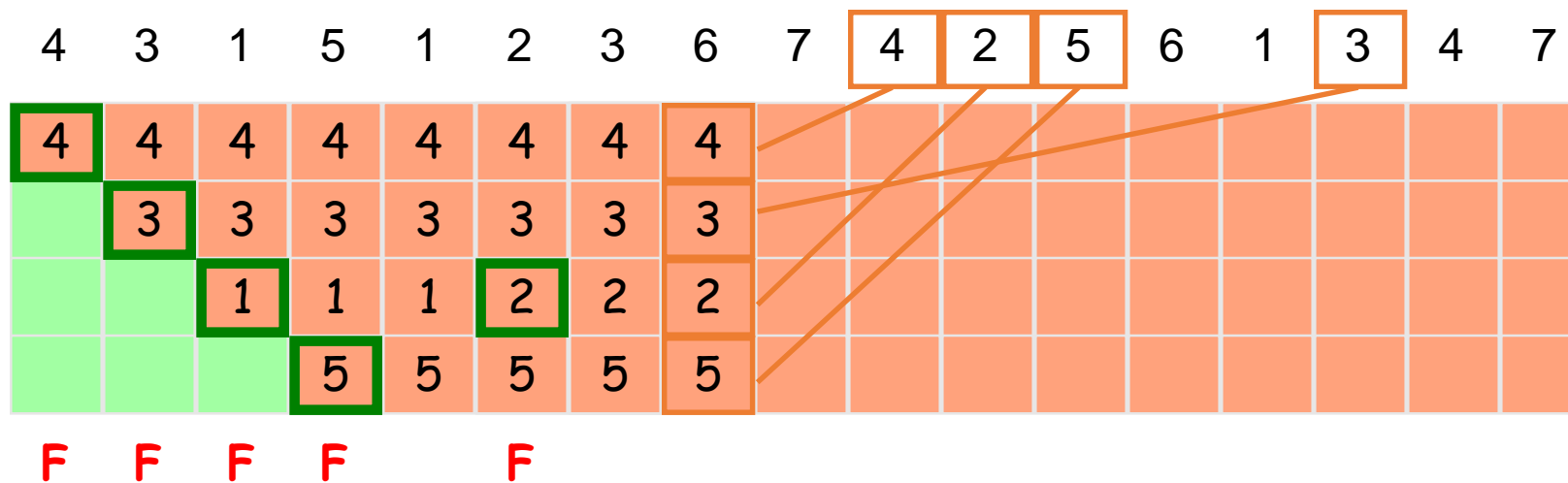
- Επτά (7) σελίδες
 - Τέσσερα (4) πλαίσια
- Ακολουθία αναφοράς



Υποψήφιες σελίδες προς αντικατάσταση: 4, 3, 1, 5
Επιλέγεται: 1

Παράδειγμα

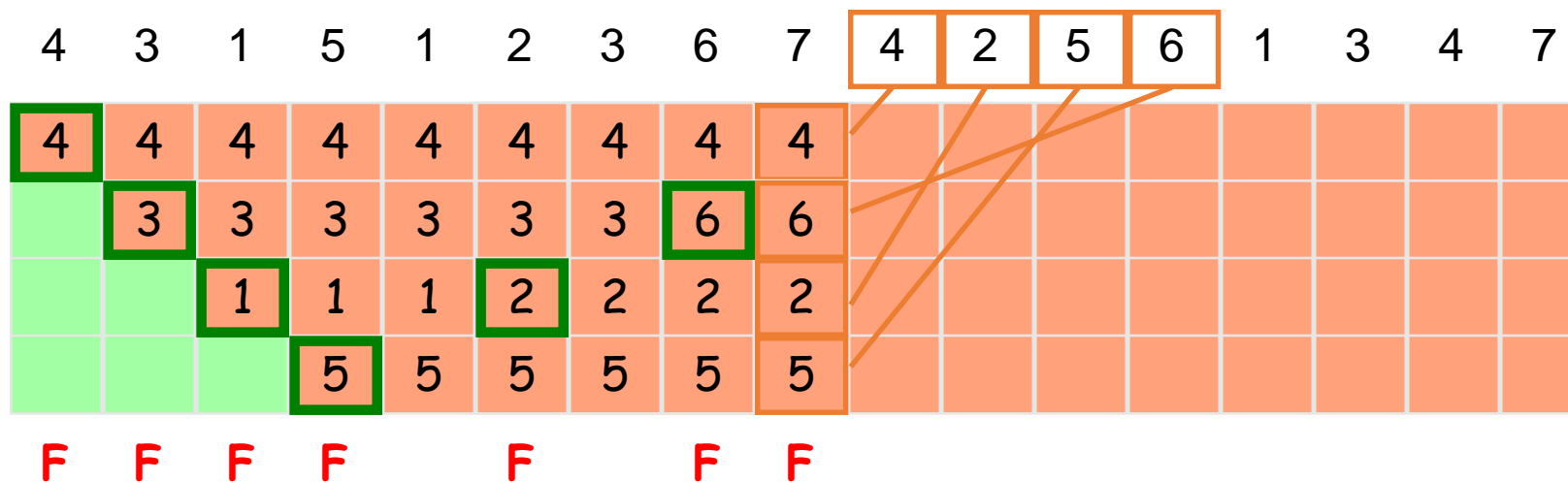
- Επτά σελίδες
 - Τέσσερα πλαίσια
- Ακολουθία αναφοράς



Υποψήφιες σελίδες προς αντικατάσταση: 4, 3, 2, 5
Επιλέγεται: 3

Παράδειγμα

- Επτά σελίδες
 - Τέσσερα πλαίσια
- Ακολουθία αναφοράς



Υποψήφιες σελίδες προς αντικατάσταση: 4, 6, 2, 5
Επιλέγεται: 6

Παράδειγμα

- Επτά σελίδες
 - Τέσσερα πλαίσια
- Ακολουθία αναφοράς

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				
	3	3	3	3	3	3	6	7	7	7	7	7				
		1	1	1	2	2	2	2	2	2	2	2				
			5	5	5	5	5	5	5	5	5	5				
F	F	F	F		F		F	F				F				

Υποψήφιες σελίδες προς αντικατάσταση: 4, 7, 2, 5

Επιλέγεται: 2 ή 5 (και οι δύο σελίδες δεν θα χρησιμοποιηθούν ξανά)

Παράδειγμα

- Επτά σελίδες
- Τέσσερα πλαίσια
Ακολουθία αναφοράς

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			
	3	3	3	3	3	3	6	7	7	7	7	7	7			
		1	1	1	2	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 προς αντικατάσταση: 4, 7, 6, 5

Επιλέγεται: 6 ή 5 (και οι δύο σελίδες δεν θα χρησιμοποιηθούν ξανά)

Παράδειγμα

- Επτά σελίδες
- Τέσσερα πλαίσια
Ακολουθία αναφοράς

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	
	3	3	3	3	3	3	6	7	7	7	7	7	7	7		
		1	1	1	2	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 προς αντικατάσταση: 4, 7, 6, 1

Επιλέγεται: 6 ή 1 (και οι δύο σελίδες δεν θα χρησιμοποιηθούν ξανά)

Παράδειγμα

- Επτά σελίδες
- Τέσσερα πλαίσια
Ακολουθία αναφοράς

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	6	6	3	3	3
			5	5	5	5	5	5	5	5	5	5	1	1	1	1
F	F	F	F		F		F	F				F	F	F		

Σφάλματα σελίδας: 10

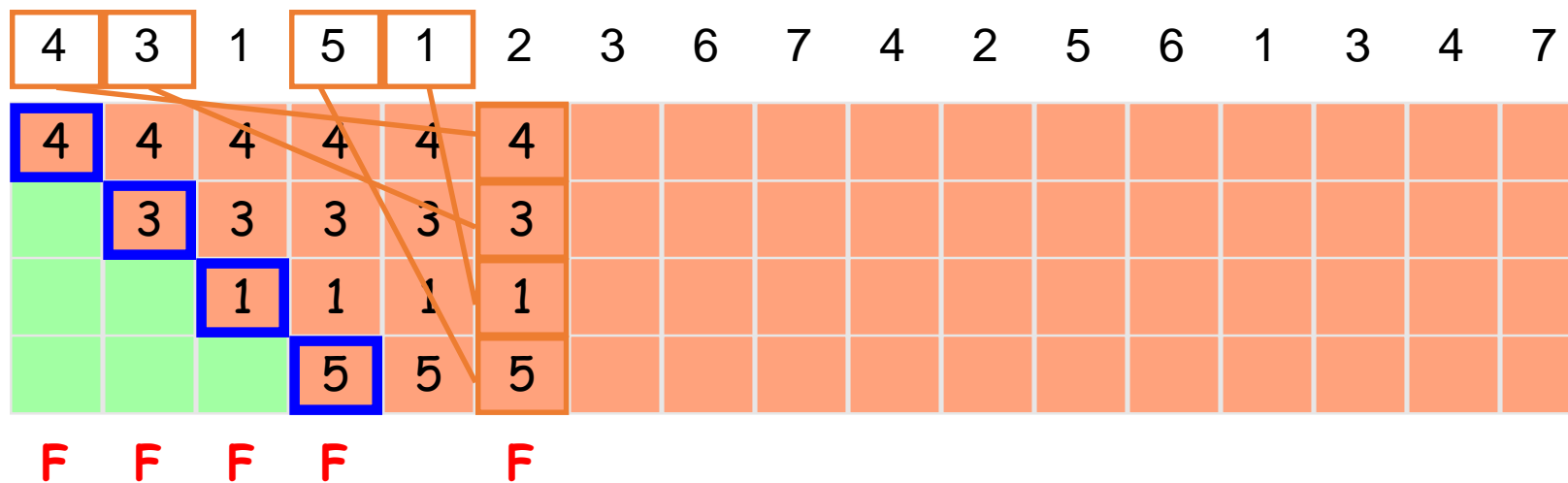
Least Recently Used (LRU)

Λιγότερο πρόσφατα χρησιμοποιούμενη σελίδα

- 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 - 3

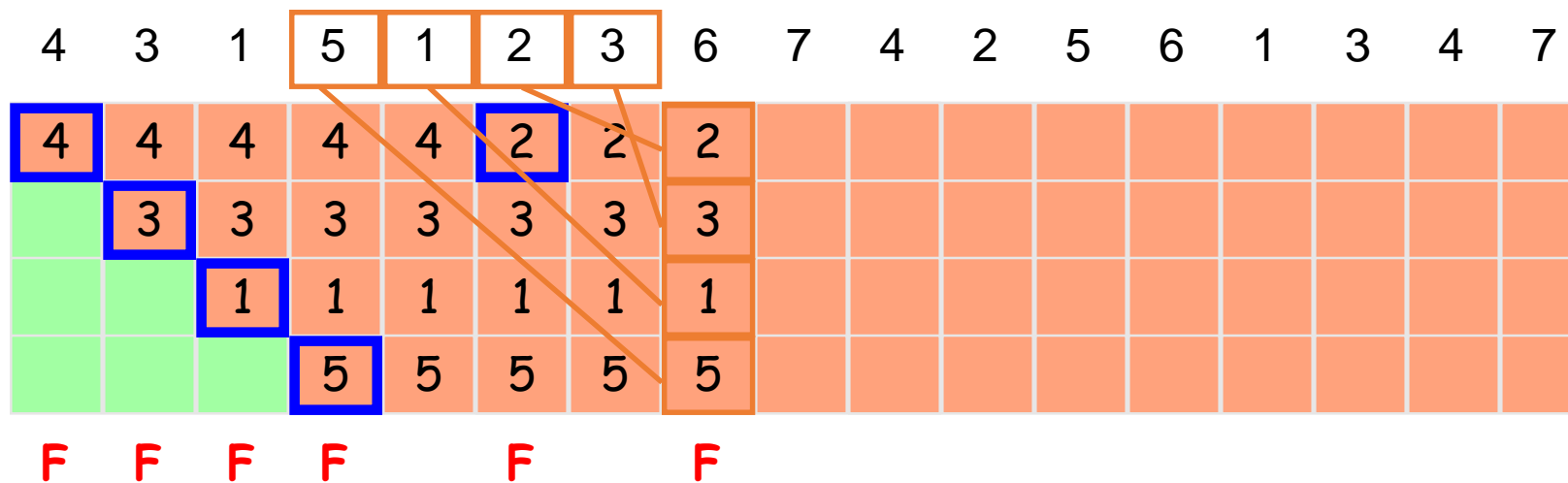
- Επτά σελίδες
 - Τέσσερα πλαίσια
- Ακολουθία αναφοράς



Υποψήφιες σελίδες προς αντικατάσταση: 4, 3, 1, 5
Επιλέγεται: 4

Παράδειγμα - 3

- Επτά σελίδες
 - Τέσσερα πλαίσια
- Ακολουθία αναφοράς



Υποψήφιες σελίδες προς αντικατάσταση: 2, 3, 1, 5
Επιλέγεται: 5

Παράδειγμα - 3

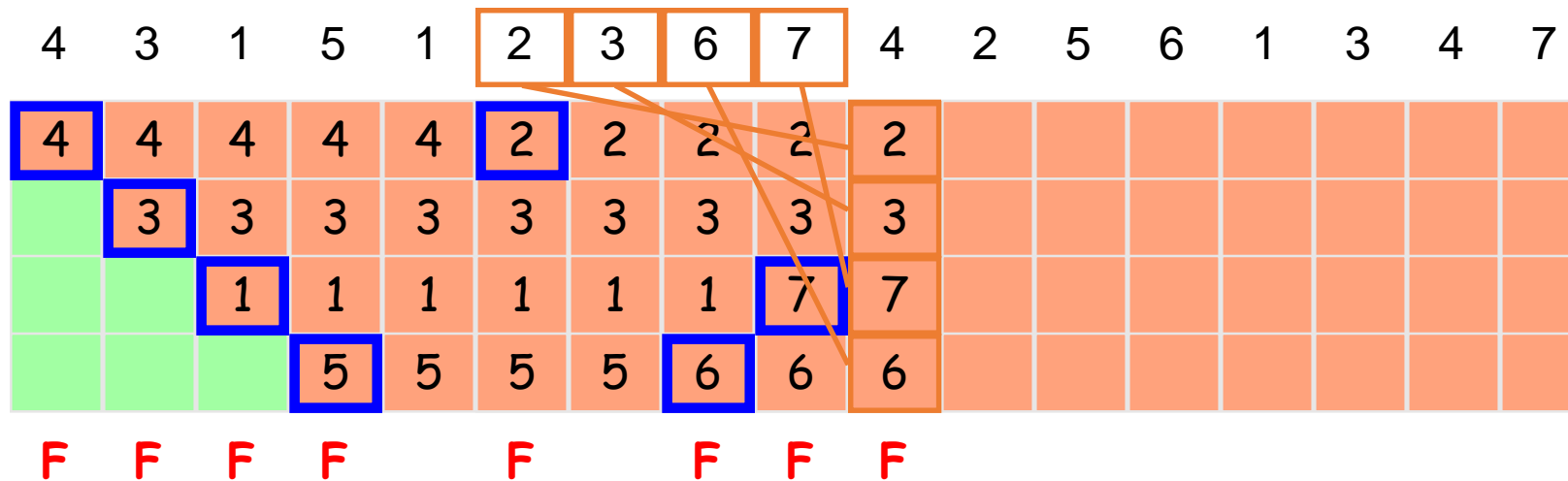
- Επτά σελίδες
 - Τέσσερα πλαίσια
- Ακολουθία αναφοράς

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 προς αντικατάσταση: 2, 3, 1, 6
Επιλέγεται: 1

Παράδειγμα - 3

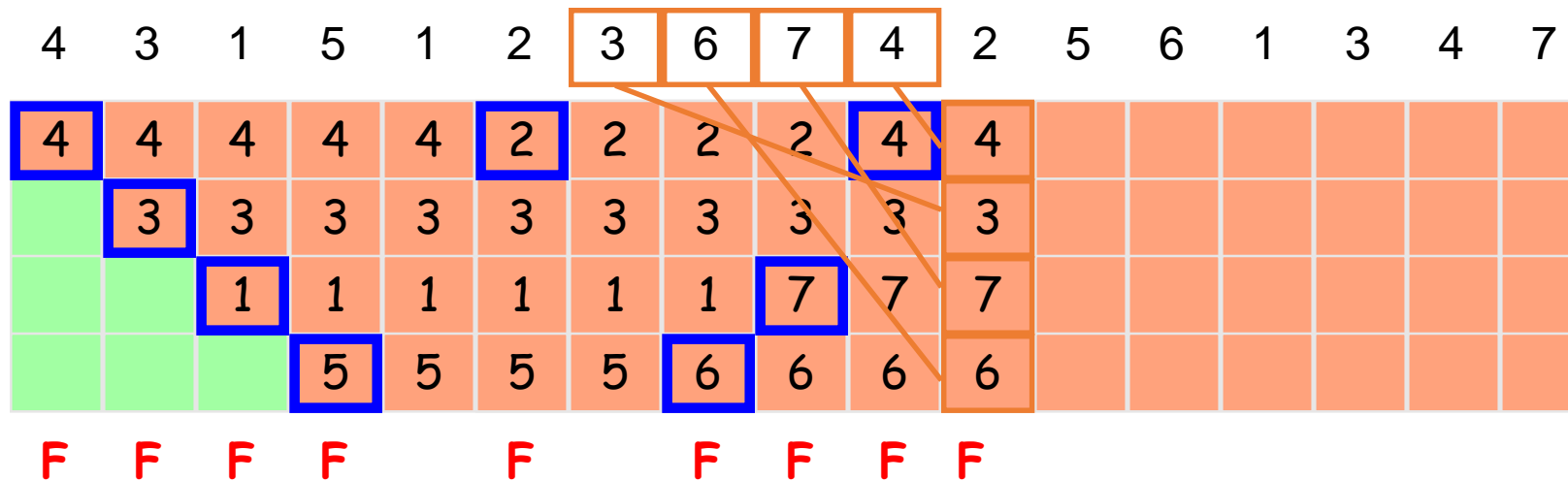
- Επτά σελίδες
 - Τέσσερα πλαίσια
- Ακολουθία αναφοράς



Υποψήφιες σελίδες προς αντικατάσταση: 2, 3, 7, 6
Επιλέγεται: 2

Παράδειγμα - 3

- Επτά σελίδες
 - Τέσσερα πλαίσια
- Ακολουθία αναφοράς



Υποψήφιες σελίδες προς αντικατάσταση: 4, 3, 7, 6
Επιλέγεται: 3

Παράδειγμα - 3

- Επτά σελίδες
- Τέσσερα πλαίσια
Ακολουθία 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 προς αντικατάσταση: 4, 2, 7, 6
Επιλέγεται: 6

Παράδειγμα - 3

- Επτά σελίδες
- Τέσσερα πλαίσια
Ακολουθία 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				
	3	3	3	3	3	3	3	3	3	2	2	2				
		1	1	1	1	1	1	7	7	7	7	7				
			5	5	5	5	6	6	6	6	5	5				
F	F	F	F		F		F	F	F	F	F	F				

Υποψήφιες σελίδες προς αντικατάσταση: 4, 2, 7, 5
Επιλέγεται: 7

Παράδειγμα - 3

- Επτά σελίδες
- Τέσσερα πλαίσια
Ακολουθία 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	4			
	3	3	3	3	3	3	3	3	3	2	2	2	2			
		1	1	1	1	1	1	7	7	7	7	6	6			
			5	5	5	5	6	6	6	6	5	5	5			
F	F	F	F		F		F	F	F	F	F	F	F			

Υποψήφιες σελίδες προς αντικατάσταση: 4, 2, 6, 5
Επιλέγεται: 4

Παράδειγμα - 3

- Επτά σελίδες
- Τέσσερα πλαίσια
Ακολουθία 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		
	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		
F	F	F	F		F		F	F	F	F	F	F	F	F		

Υποψήφιες σελίδες προς αντικατάσταση: 1, 2, 6, 5
Επιλέγεται: 2

Παράδειγμα - 3

- Επτά σελίδες
- Τέσσερα πλαίσια
Ακολουθία 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	6	
			5	5	5	5	6	6	6	6	5	5	5	5	5	
F	F	F	F		F		F	F	F	F	F	F	F	F	F	

Υποψήφιες σελίδες προς αντικατάσταση: 1, 3, 6, 5
Επιλέγεται: 5

Παράδειγμα - 3

- Επτά σελίδες
- Τέσσερα πλαίσια
Ακολουθία 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	3	2	2	2	2	3	3	3
		1	1	1	1	1	1	7	7	7	7	6	6	6	6	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	F

Υποψήφιες σελίδες προς αντικατάσταση: 1, 3, 6, 4
Επιλέγεται: 6

Παράδειγμα - 3

- Επτά σελίδες
 - Τέσσερα πλαίσια
- Ακολουθία 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	3	2	2	2	2	3	3	3
		1	1	1	1	1	1	7	7	7	7	6	6	6	6	7
			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	F

Σφάλματα σελίδας: 15