Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής

Αρχιτεκτονική Υπολογιστών

2022-23

Εικονική Μνήμη

(και ο ρόλος της στην ιεραρχία μνήμης)

http://mixstef.github.io/courses/comparch/

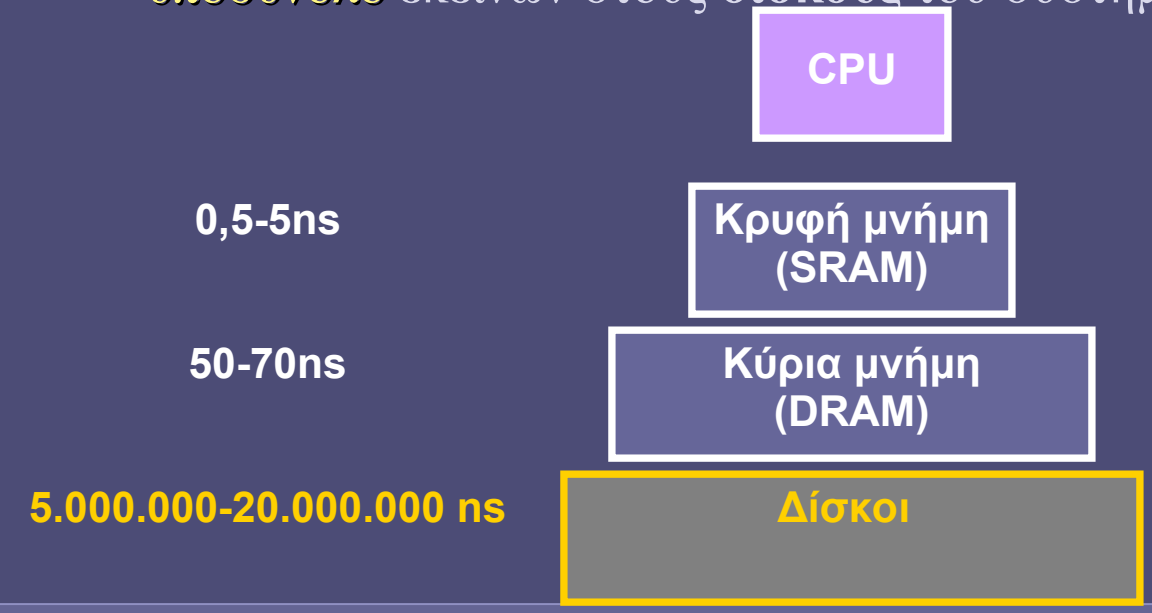
Μ.Στεφανιδάκης

Επεκτείνοντας την Ιεραρχία Μνήμης

- Ιεραρχία Μνήμης

Επέκταση ιεραρχίας εκτός του συστήματος: δικτυακές θέσεις αποθήκευσης

- Εννοιολογικά:
  - Κάθε υψηλότερο επίπεδο δρα ως «κρυφή μνήμη» για το αμέσως χαμηλότερο
    - Η **κύρια μνήμη** λειτουργεί ως «κρυφή μνήμη» των **δίσκων**
    - Τα περιεχόμενα στην **κύρια μνήμη** είναι υποσύνολο εκείνων στους **δίσκους** του συστήματος



Εικονική μνήμη (virtual memory)

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

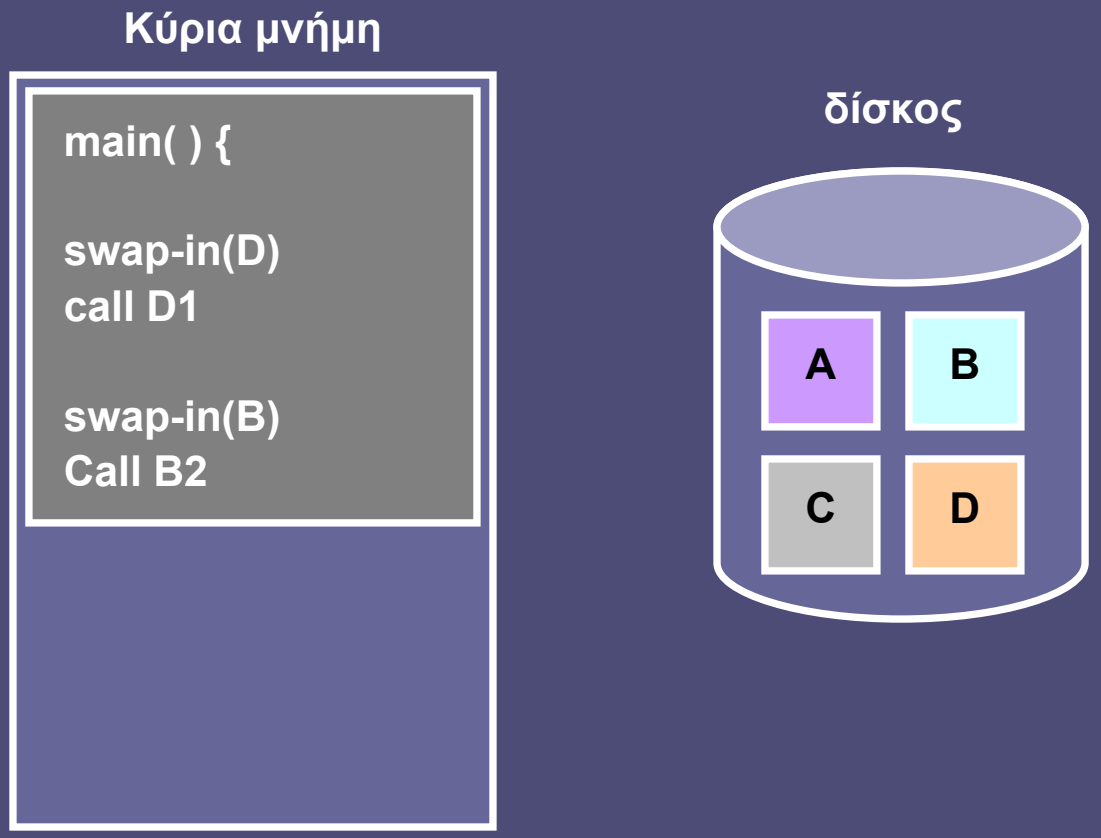
Το πρώτο σύστημα εικονικής μνήμης παρουσιάστηκε το 1962 (Atlas computer)

- Για ποιον λόγο εμφανίστηκε;
  - Στους πρώτους υπολογιστές το μέγεθος της κύριας μνήμης ήταν περιορισμένο
  - Ακόμα και στην περίπτωση του μονοπρογραμματισμού η κύρια μνήμη ήταν ανεπαρκής
  - Εμφάνιση ΛΣ με υποστήριξη πολυπρογραμματισμού
    - Αδυναμία ταυτόχρονης διατήρησης πολλών προγραμμάτων στην κύρια μνήμη
  - Η λύση: εικονική μνήμη
    - Μέρος των δεδομένων βρίσκεται στους δίσκους του συστήματος
    - Μεταφορά στην κύρια μνήμη όταν χρειαστεί
    - Πιθανότατα αντικαθιστώντας άλλα τμήματα δεδομένων
    - Τα τελευταία μεταφέρονται πίσω στους δίσκους

Πριν την εικονική μνήμη: overlays

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η τεχνική των overlays απαιτούσε από τον προγραμματιστή να χειρίζεται τις λεπτομέρειες (και διευθύνσεις) φόρτωσης και κλήσης των υπο-προγραμμάτων!

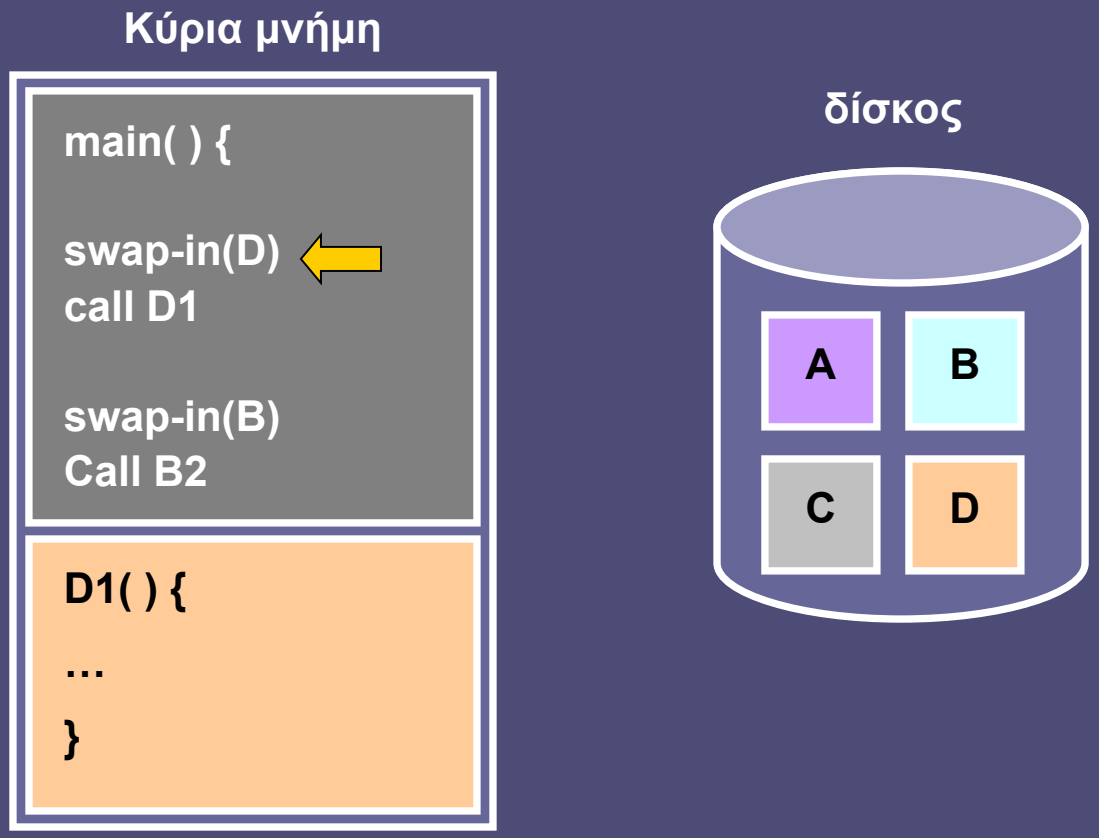


- Καταλληλότερο για στατικά δεδομένα
  - όπως ο κώδικας των υποπρογραμμάτων

Πριν την εικονική μνήμη: overlays

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

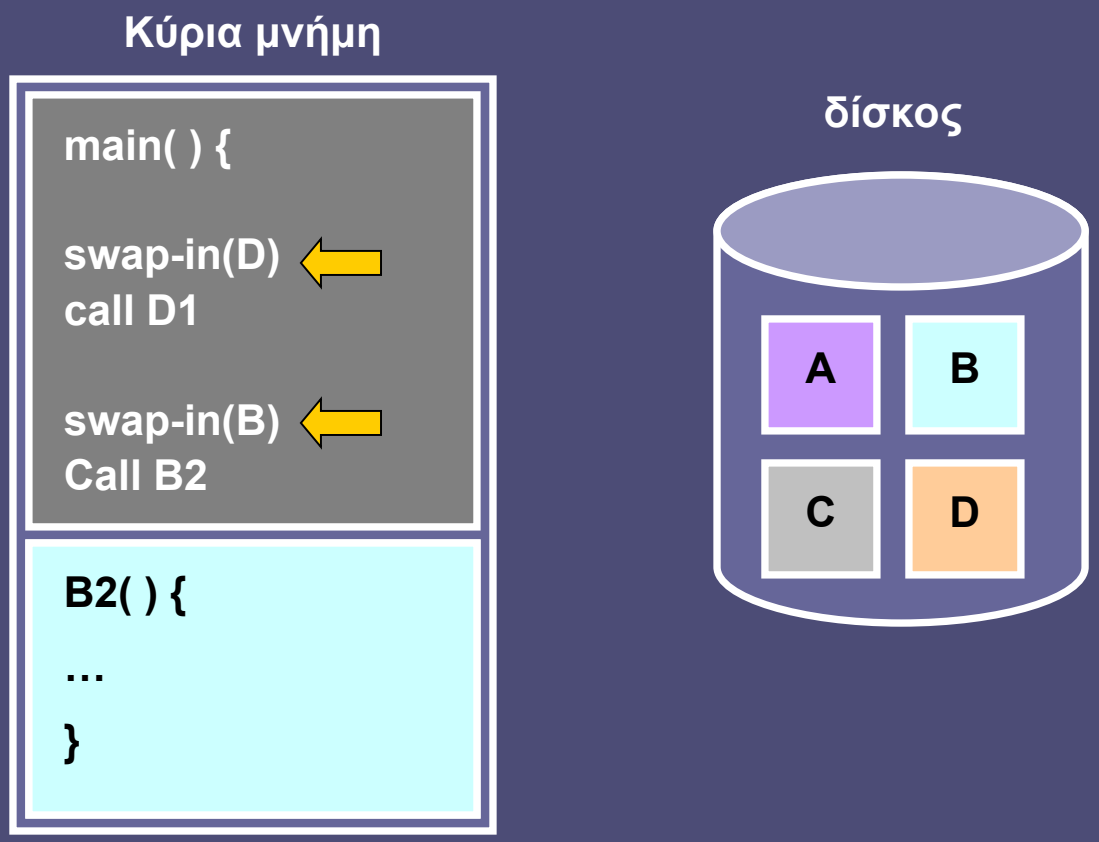
Η τεχνική των overlays απαιτούσε από τον προγραμματιστή να χειρίζεται τις λεπτομέρειες (και διευθύνσεις) φόρτωσης και κλήσης των υπο-προγραμμάτων!



Πριν την εικονική μνήμη: overlays

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η τεχνική των overlays απαιτούσε από τον προγραμματιστή να χειρίζεται τις λεπτομέρειες (και διευθύνσεις) φόρτωσης και κλήσης των υπο-προγραμμάτων!



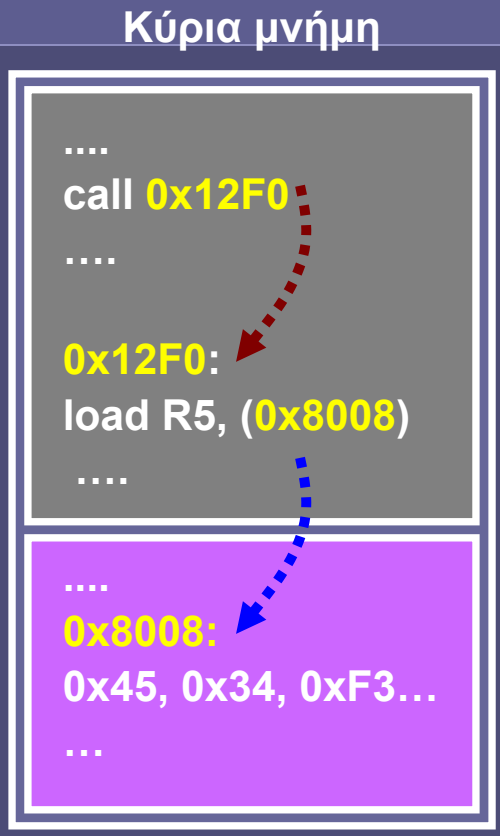
Χώρος διευθύνσεων προγράμματος

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Ποιες οι διευθύνσεις που δημιουργούσε ο μεταγλωττιστής;

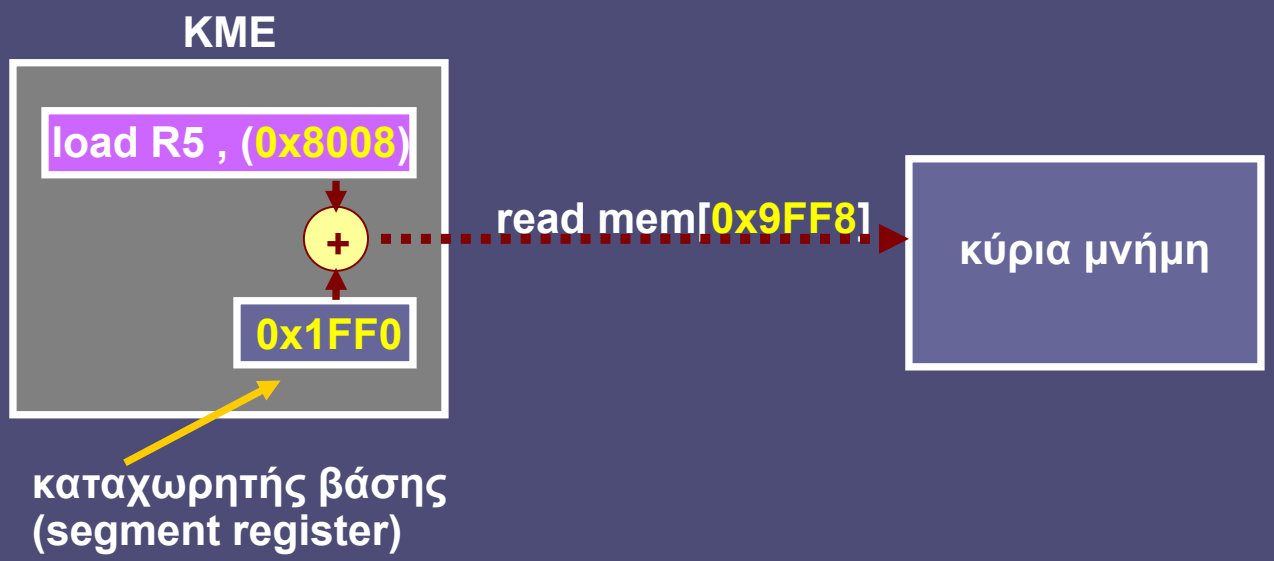
Τι συνέβαινε αν το πρόγραμμα δεν φορτωνόταν πάντοτε στον ίδιο χώρο μνήμης;

- Address Space
  - Εκτελούμενο πρόγραμμα στη μνήμη:
  - Διευθύνσεις κώδικα
    - Εντολές διακλάδωσης
  - Διευθύνσεις δεδομένων
    - Εντολές load-store
- Πριν την εικονική μνήμη:
  - Κάθε πρόγραμμα χρησιμοποιούσε φυσικές διευθύνσεις της κύριας μνήμης



Πολυπρογραμματισμός πριν την εικονική μνήμη

- Ιεραρχία Μνήμης
- Εικονική Μνήμη



- Ειδικοί καταχωρητές βάσης για κώδικα και δεδομένα
  - Διευκόλυνση τοποθέτησης προγραμμάτων οπουδήποτε στη μνήμη
  - Αλλαγή τιμής καταχωρητών βάσης ανά πρόγραμμα

Μεταβαίνοντας σε λογικές διευθύνσεις

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

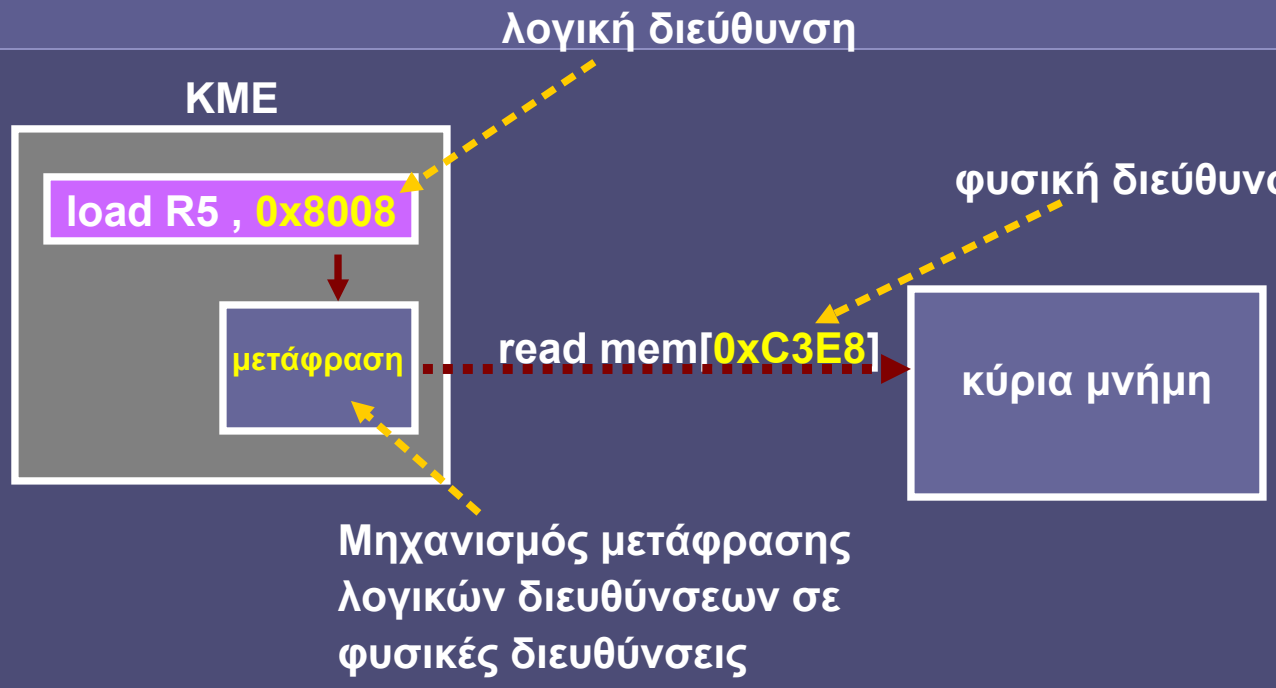
- Το προηγούμενο σχήμα
  - Εισήγαγε την αποσύνδεση των λογικών (εικονικών) διευθύνσεων των προγραμμάτων από τις φυσικές διευθύνσεις κύριας μνήμης
  - Με απλή αντιστοιχία:

    φυσική διεύθυνση = λογική διεύθυνση + καταχωρητής βάσης
  - Απαιτείται υποστήριξη από το υλικό (ΚΜΕ)
  - Το πρόγραμμα μπορεί να φορτωθεί σε οποιαδήποτε θέση μνήμης (relocation)
    - Δεν περιέχει αναφορές σε φυσικές διευθύνσεις
  - Εισάγεται η έννοια των ξεχωριστών χώρων διευθύνσεων (κώδικα, δεδομένων...) ανά πρόγραμμα
    - χωρίς περαιτέρω υποστήριξη

Λογικές (εικονικές) διευθύνσεις

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Υπάρχουν «μικροί» επεξεργαστές (μικροελεγκτές – microcontrollers, MCUs) που δεν χρειάζεται να παρέχουν μηχανισμό εικονικής μνήμης



- Για τη μετάφραση των λογικών διευθύνσεων σε φυσικές απαιτείται συνδυασμένη υποστήριξη από KME, κρυφή μνήμη και λειτουργικό σύστημα

Σύστημα εικονικής μνήμης

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

- Ποιος ο ρόλος ενός συστήματος εικονικής μνήμης;
  - Η δυνατότητα χρήσης εικονικής μνήμης πέρα από όση είναι πραγματικά διαθέσιμη και η διευκόλυνση του πολυπρογραμματισμού
    - Ο αρχικός λόγος για τη χρήση της εικονικής μνήμης
  - Η υλοποίηση μηχανισμών προνομίων προσπέλασης μνήμης από κάθε διεργασία και η προστασία των δεδομένων μεταξύ διαφορετικών διεργασιών
    - Σημαντικότερο σήμερα, ιδίως στο cloud computing!
- Ποιος διαχειρίζεται την εικονική μνήμη;
  - Διαχείριση από το λειτουργικό σύστημα
  - Υποστήριξη από το υλικό (ΚΜΕ/κρυφή μνήμη)

Υλοποίηση εικονικής μνήμης

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

- Εκμετάλλευση της αρχής της τοπικότητας
  - Μερικά μέρη μόνο των προγραμμάτων είναι «ενεργά» κάθε στιγμή
    - Χρησιμοποιώντας μικρό μέρος από τον συνολικό χώρο διευθύνσεών τους (κώδικα και δεδομένα)
  - Το λειτουργικό σύστημα αφαιρεί από την κύρια μνήμη και τοποθετεί στους δίσκους τα τμήματα μνήμης που δεν έχουν χρησιμοποιηθεί στο πρόσφατο παρελθόν
    - Συνεπώς θα χρειαστούν στο άμεσο μέλλον με μικρή πιθανότητα
    - Η διαχείριση της εικονικής μνήμης έχει ομοιότητες με τη διαχείριση κρυφής-κύριας μνήμης
      - Η διαφορά στο κόστος προσπέλασης όμως μεταξύ κύριας μνήμης – δίσκων είναι πολύ μεγαλύτερη (100.000 φορές και πλέον)

Σελιδοποίηση (paging)

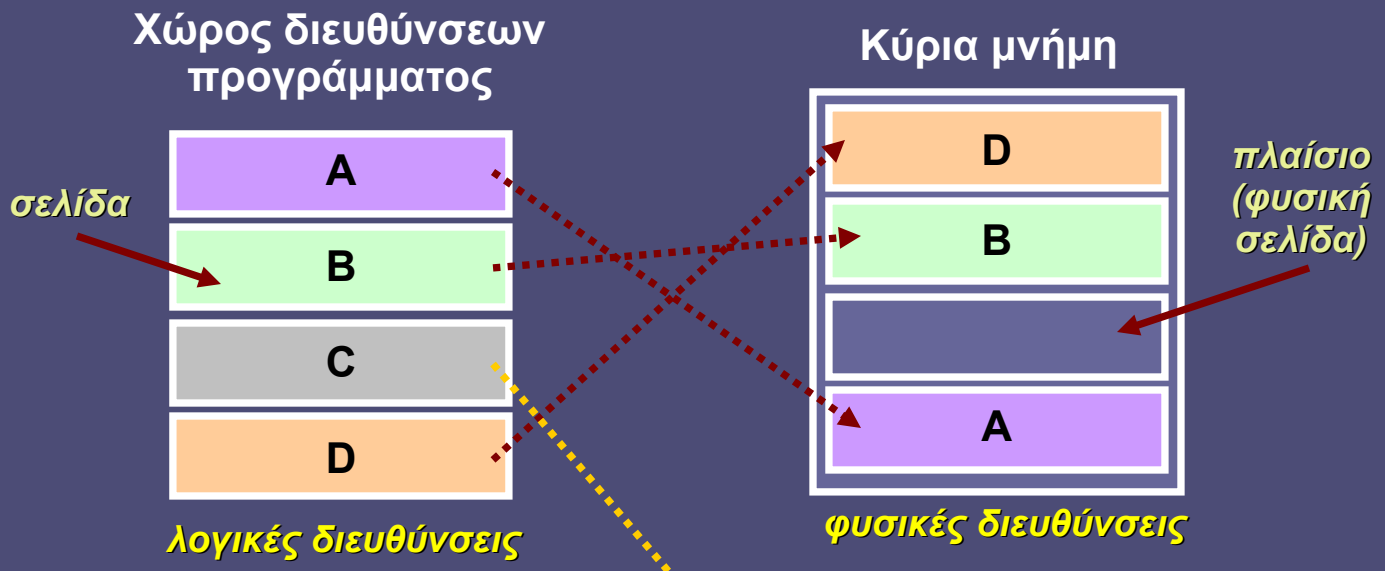
- Ιεραρχία Μνήμης
- Εικονική Μνήμη

- Ο χώρος λογικών διευθύνσεων κάθε προγράμματος χωρίζεται σε σελίδες (pages)
  - Συγκεκριμένα μεγέθη σελίδας (4KB, 2MB, 1GB)
    - 4KB είναι το «κλασσικό μέγεθος», ταιριάζει με το μπλοκ μεταφοράς δεδομένων από/προς δίσκους
      - Μεγαλύτερο μέγεθος (=λιγότερες σελίδες ανά πρόγραμμα) → λιγότεροι πόροι για διαχείριση
      - Αλλά και αύξηση του αχρησιμοποίητου χώρου
- Η κύρια μνήμη (φυσικές διευθύνσεις) χωρίζεται σε φυσικές σελίδες (ή πλαίσια, frames) με μέγεθος ίσο με της σελίδας
  - Κάθε σελίδα τοποθετείται σε ένα ελεύθερο πλαίσιο
  - Ευκολία τοποθέτησης και αντικατάστασης σελίδων στην κύρια μνήμη

Βασικό σχήμα σελιδοποίησης

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Σύμφωνα με τον μηχανισμό της εικονικής μνήμης μια σελίδα μπορεί να βρίσκεται στον δίσκο



- Χώρος διευθύνσεων προγράμματος
  - Σελίδες (pages)
- Φυσικές σελίδες στη μνήμη
  - Πλαίσια (frames)
  - Ίδιο μέγεθος με σελίδες

Σελιδοποίηση κατ'απαίτηση (demand paging)

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

- Οι σελίδες των προγραμμάτων (κώδικας-δεδομένα) βρίσκονται αρχικά μόνο στον δίσκο
  - Το ΛΣ τις σημειώνει ως "απούσες" από τη μνήμη
- Όταν η εκτελούμενη εντολή προσπελάσει δεδομένα από μια "απούσα" σελίδα, δημιουργείται ένα σφάλμα σελιδοποίησης (page fault)...
- ...και το ΛΣ τη φορτώνει σε ένα πλαίσιο στη μνήμη
  - Ενδεχομένως εκτοπίζοντας πίσω στον δίσκο μια άλλη σελίδα από τη μνήμη
    - Η τελευταία σημειώνεται ως "απούσα"
- Στη συνέχεια το λειτουργικό επιστρέφει τον έλεγχο στο πρόγραμμα που προκάλεσε το σφάλμα σελιδοποίησης
  - Η εντολή που διακόπηκε εκτελείται ξανά

Κρίσιμα σημεία στη σχεδίαση εικονικής μνήμης

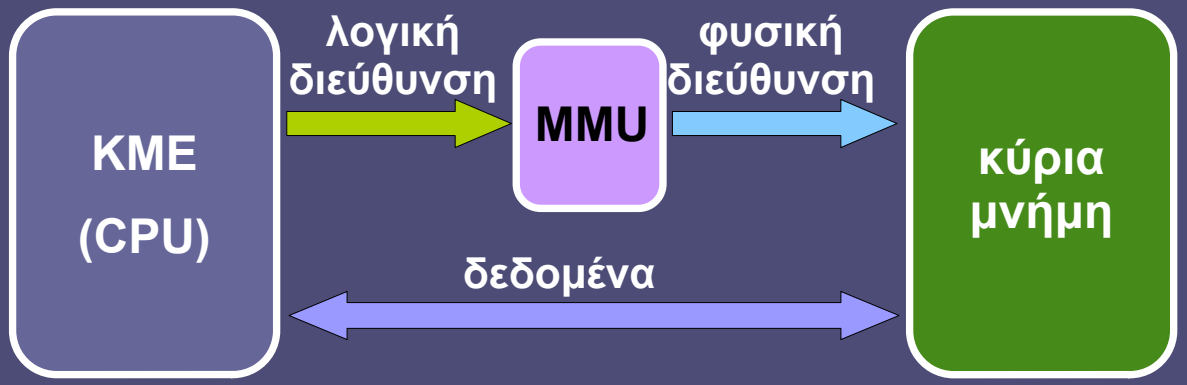
- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Ακόμα και μικρή μείωση στην εμφάνιση page faults μπορεί να έχει σημαντικό όφελος για την απόδοση του συστήματος

- Η μείωση των page faults είναι επιβεβλημένη
  - Page faults: μεγάλο κόστος σε κύκλους αναμονής (1-10Μκύκλοι)
  - Οι σελίδες τοποθετούνται οπουδήποτε μέσα στη μνήμη
    - Σχήμα ανάλογο των fully-associative κρυφών μνημών
- Οι σελίδες πρέπει να έχουν ικανό μέγεθος για εξισορρόπηση του κόστους προσπέλασης του δίσκου
  - Δεν είναι δυνατή η ενημέρωση στον δίσκο με κάθε εγγραφή νέων δεδομένων στη σελίδα
- Η διαχείριση της εικονικής μνήμης γίνεται από το λειτουργικό σύστημα
  - Μικρή επιβάρυνση συγκρινόμενη με χρόνο μετακίνησης σελίδων στους δίσκους
  - Δυνατότητα χρήσης πολυπλοκότερων αλγορίθμων για τοποθέτηση-αντικατάσταση σελίδων στη μνήμη

Εικονική μνήμη: το υλικό

- Ιεραρχία Μνήμης
- Εικονική Μνήμη



- Σύστημα διαχείρισης μνήμης
  - Memory management unit – MMU
  - Βρίσκεται μέσα στον επεξεργαστή
  - Μεταφράζει τις λογικές (εικονικές) διευθύνσεις κάθε προγράμματος σε φυσικές διευθύνσεις μνήμης

Από το διπλανό σχήμα απουσιάζει η κρυφή μνήμη, η οποία συνδέεται στενά με τον μηχανισμό εικονικής μνήμης (θα το δούμε αργότερα)

Σελίδες και λογικές (εικονικές) διευθύνσεις

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η μετάφραση γίνεται στον επεξεργαστή (MMU), άρα εκεί (δηλ. στο υλικό) καθορίζεται το μέγεθος της σελίδας

Ο επεξεργαστής παράγει λογική (εικονική) διεύθυνση N bits

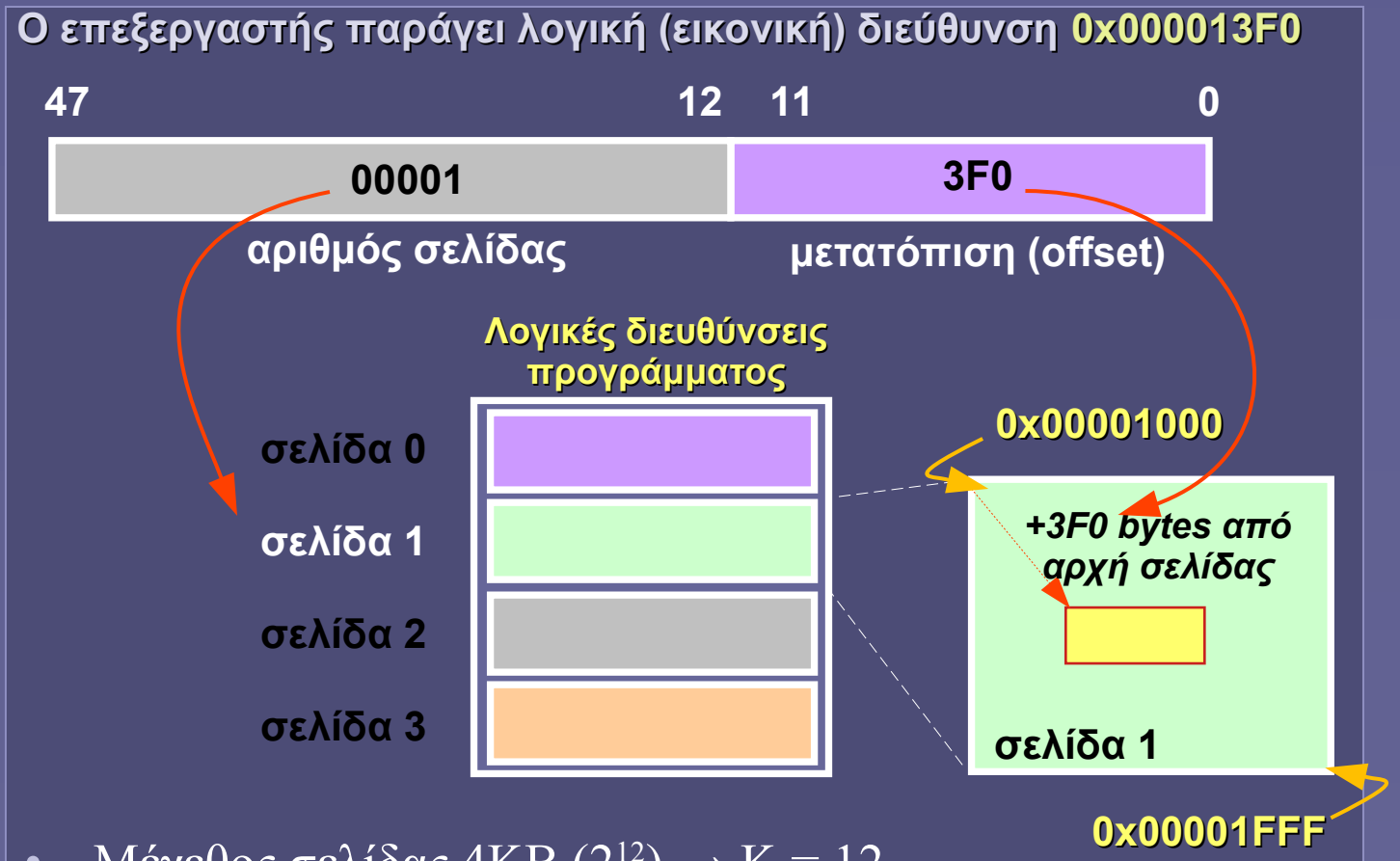
Στη μνήμη στέλνεται φυσική διεύθυνση M bits

- Ο επεξεργαστής μπορεί να παράγει έως και 2^N εικονικές διευθύνσεις
- Η φυσική μνήμη μπορεί να έχει έως 2^M διευθύνσεις
- Το μέγεθος σελίδας είναι 2^K bytes
- Τα N και M μπορούν να έχουν οποιαδήποτε σχέση

Παράδειγμα με μέγεθος σελίδας 4KB

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η μετάφραση γίνεται στον επεξεργαστή (MMU), άρα εκεί (δηλ. στο υλικό) καθορίζεται το μέγεθος της σελίδας

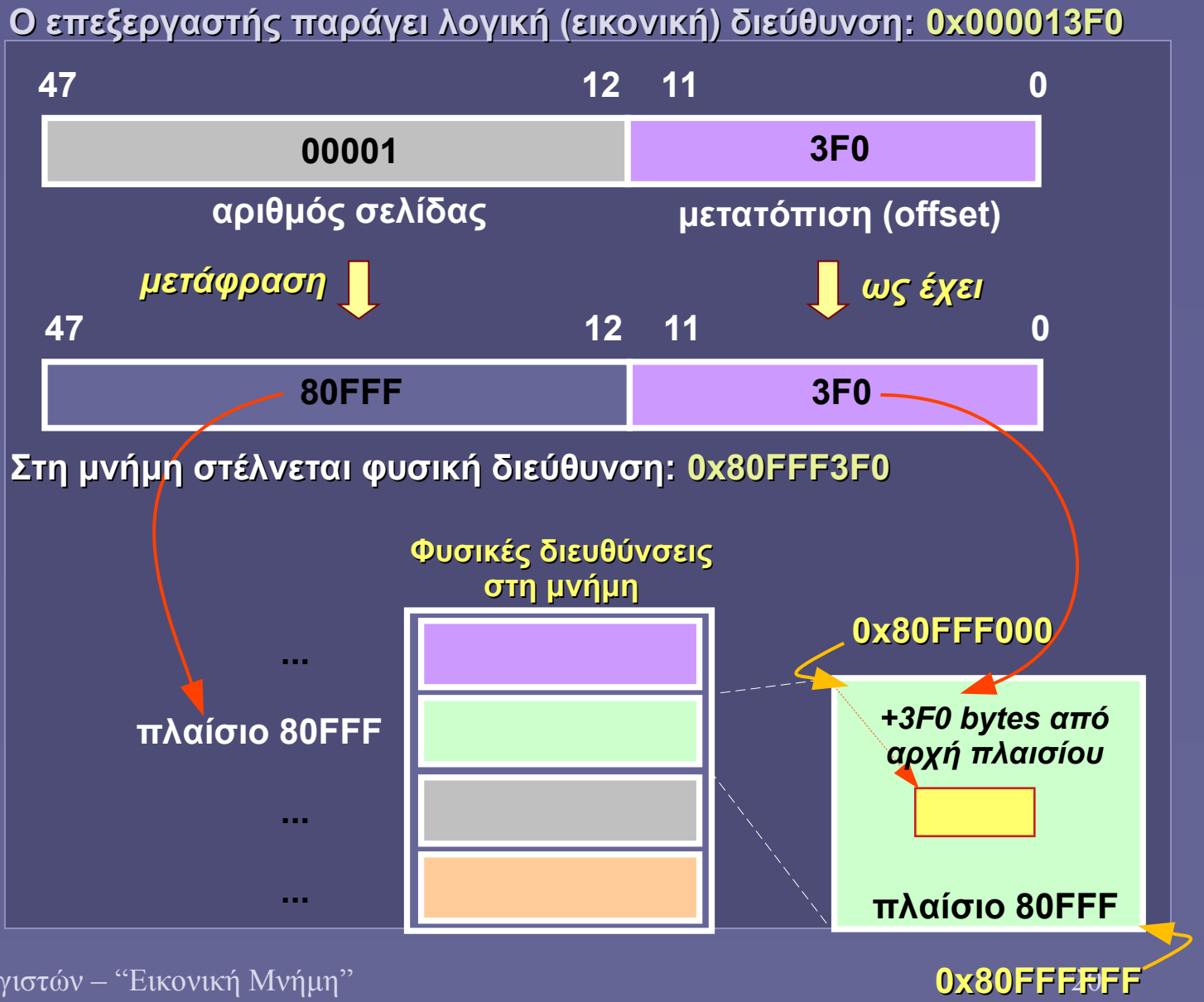


- Μέγεθος σελίδας 4KB (2^{12}) → K = 12
- Έστω ότι λογικές και φυσικές διευθύνσεις έχουν εύρος 48 bits → N = M = 48

Παράδειγμα (συνέχεια)

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Έστω ότι η σελίδα 0x00001 βρίσκεται στη φυσική σελίδα (πλαίσιο) 0x80FFF



Πίνακας σελίδων

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

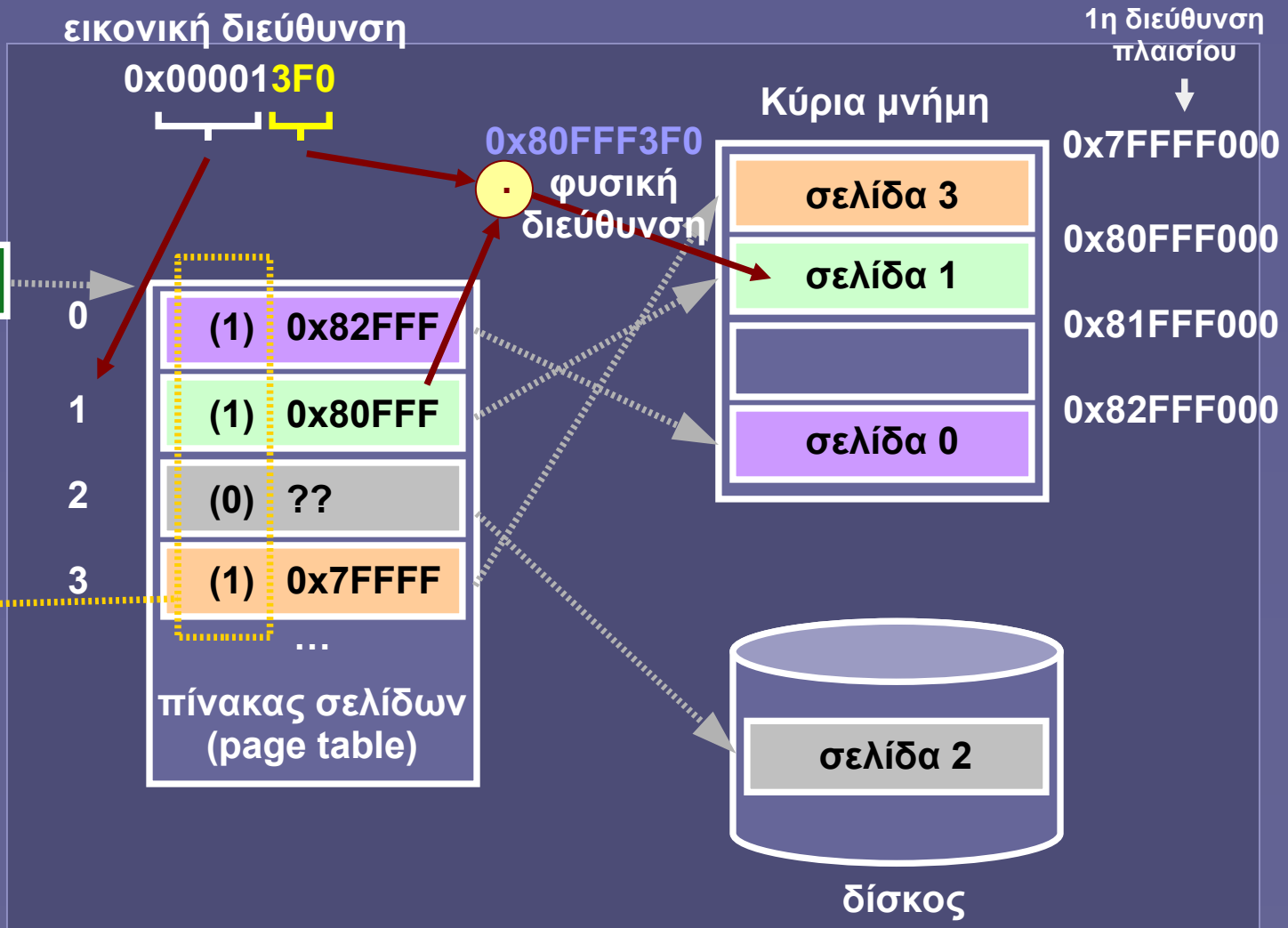
- Η μονάδα διαχείρισης μνήμης (MMU) χρειάζεται τις αντιστοιχίες σελίδων και πλαισίων για τη μετάφραση από λογικές σε φυσικές διευθύνσεις
  - Στο προηγούμενο παράδειγμα: πώς γνωρίζει η μονάδα MMU ότι η σελίδα 0x00001 βρίσκεται στη φυσική σελίδα (πλαίσιο) 0x80FFF;
- Οι αντιστοιχίες αυτές αποθηκεύονται σε έναν πίνακα σελίδων (page table)
  - Βρίσκεται στην κύρια μνήμη
    - Page table register: διεύθυνση αρχής του πίνακα
  - Έχει μέγεθος (θεωρητικά) ίσο με τον μέγιστο αριθμό σελίδων ενός προγράμματος
    - Μια γραμμή ανά σελίδα
    - Π.χ για 2^{20} σελίδες των 4KB, με 4 bytes ανά γραμμή του πίνακα, απαιτούνται 4MB
  - Διαχείριση από το λειτουργικό σύστημα

Μετάφραση λογικών διευθύνσεων

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

page table register

Presence (ή valid) bit: βρίσκεται η σελίδα στην κύρια μνήμη ή στον δίσκο;



Πίνακας σελίδων (συνέχεια)

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

- **Πρόσθετη πληροφορία σε κάθε γραμμή του πίνακα σελίδων**
  - Presence (ή valid) bit
    - Βρίσκεται η σελίδα στη μνήμη; ή στο δίσκο
  - Dirty bit
    - Έχει γίνει εγγραφή δεδομένων στη σελίδα;
  - Accessed (ή reference ή use) bit
    - Ενημερώνεται από MMU σε κάθε προσπέλαση
    - Καταγράφεται σε τακτά διαστήματα από το ΛΣ για την υλοποίηση των αλγορίθμων LRU κατά την αντικατάσταση σελίδων
  - Μέγεθος σελίδας
    - Αν υποστηρίζονται εναλλακτικά μεγέθη
  - Δικαιώματα χρήσης σελίδας
    - read only/read write, user/supervisor …
    - Ανάλογα με αρχιτεκτονική επεξεργαστή

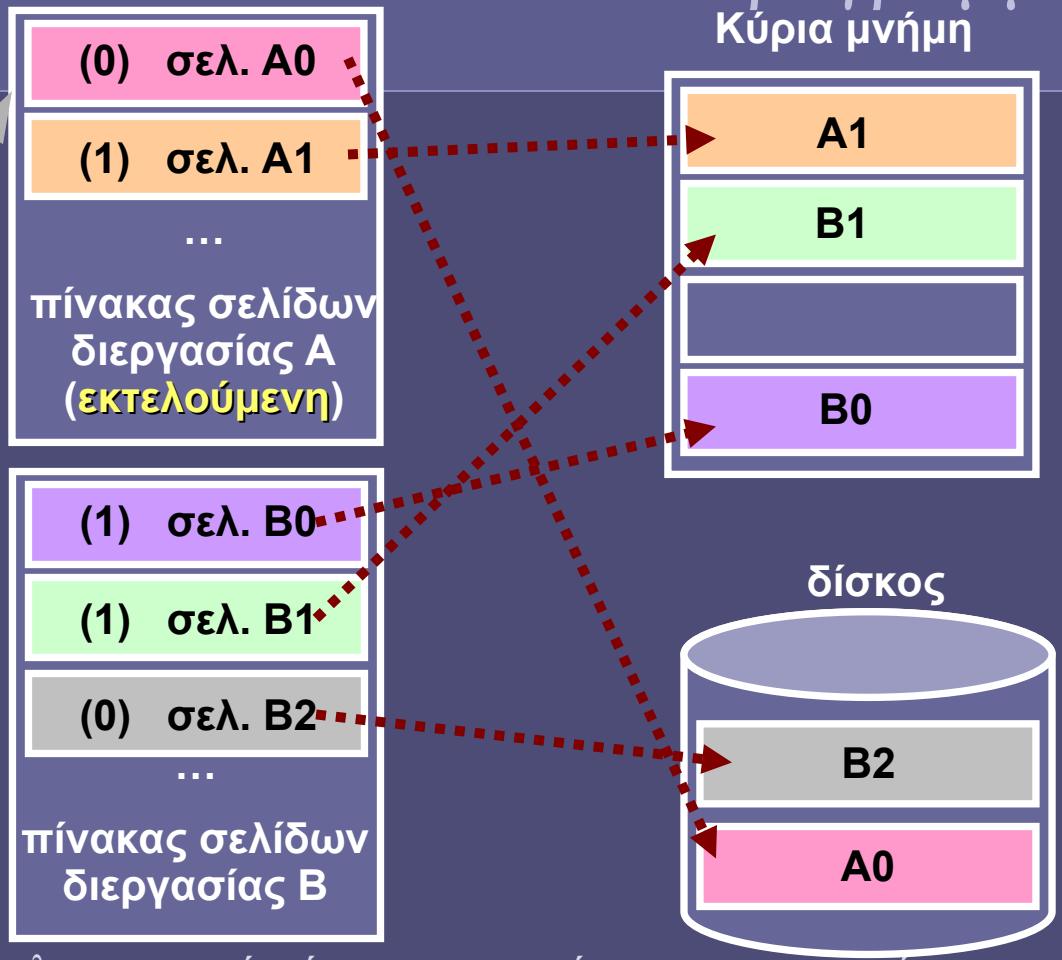
Πίνακες σελίδων και πολλαπλά προγράμματα

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

page table register

Κάθε διεργασία έχει τον δικό της πίνακα σελίδων

Κατά την εναλλαγή διεργασιών αλλάζει και ο καταχωρητής-δείκτης στιν πίνακα σελίδων (page table register)



Εάν το λειτουργικό σύστημα φροντίσει να αντιστοιχίσει σωστά τις σελίδες κάθε προγράμματος σε διαφορετικές διευθύνσεις στην κύρια μνήμη, τότε κανένα πρόγραμμα δεν μπορεί να προσπελάσει δεδομένα άλλου προγράμματος

Εικονική Μνήμη και Προστασία Προσπέλασης

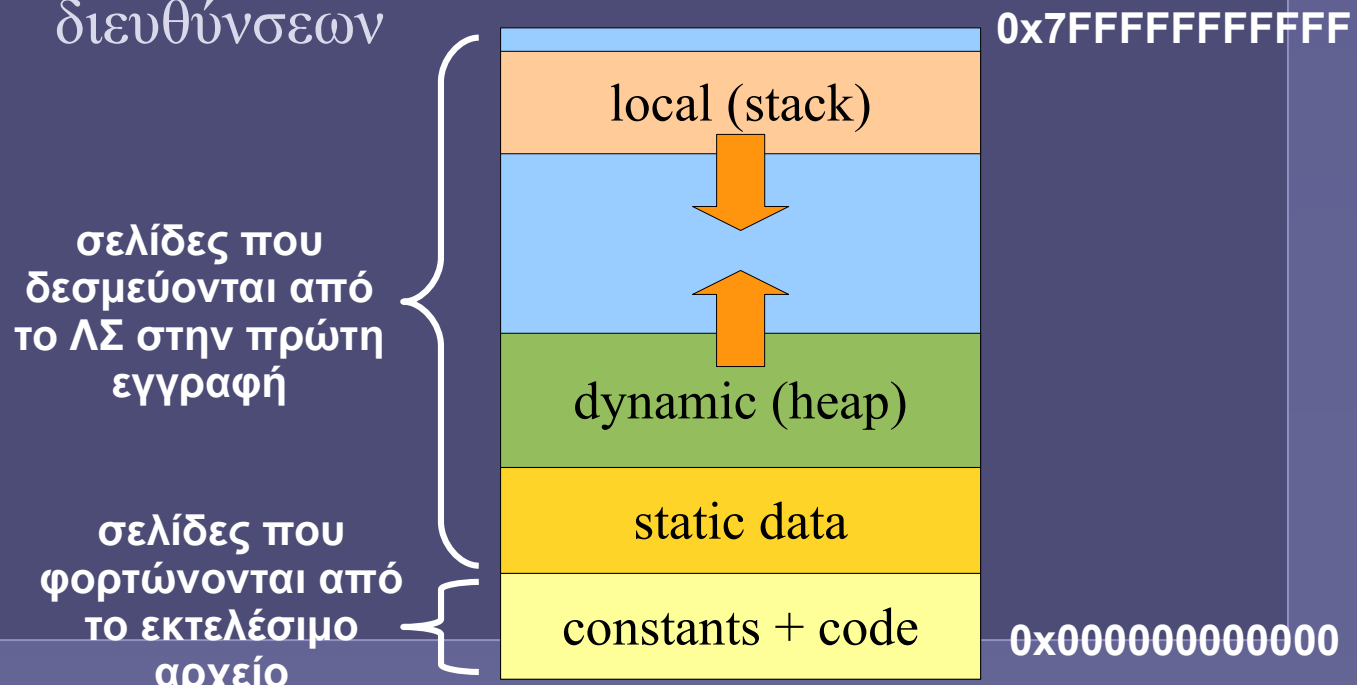
- Προστασία προσπέλασης σελίδων
  - Με διαφορετικούς πίνακες σελίδων ανά πρόγραμμα (διεργασία) είναι αδύνατη η προσπέλαση «ξένων» σελίδων
- User mode και Supervisor Mode
  - Σε user mode δεν είναι δυνατή η προσπέλαση του πίνακα σελίδων και των αντίστοιχων καταχωρητών
    - Υπάρχουν αρχιτεκτονικές με περισσότερα από 2 επίπεδα προνομίων
- Ελεγχόμενη προσπέλαση κώδικα ΛΣ
  - System call exceptions
    - Στο επίπεδο όμως των προνομίων του χρήστη → δεν είναι δυνατή η προσπέλαση «ξένων» δεδομένων

Μέγεθος πίνακα σελίδων

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Δεν είναι δυνατό να έχουμε πίνακα σελίδων με γραμμές για κάθε πιθανή σελίδα.

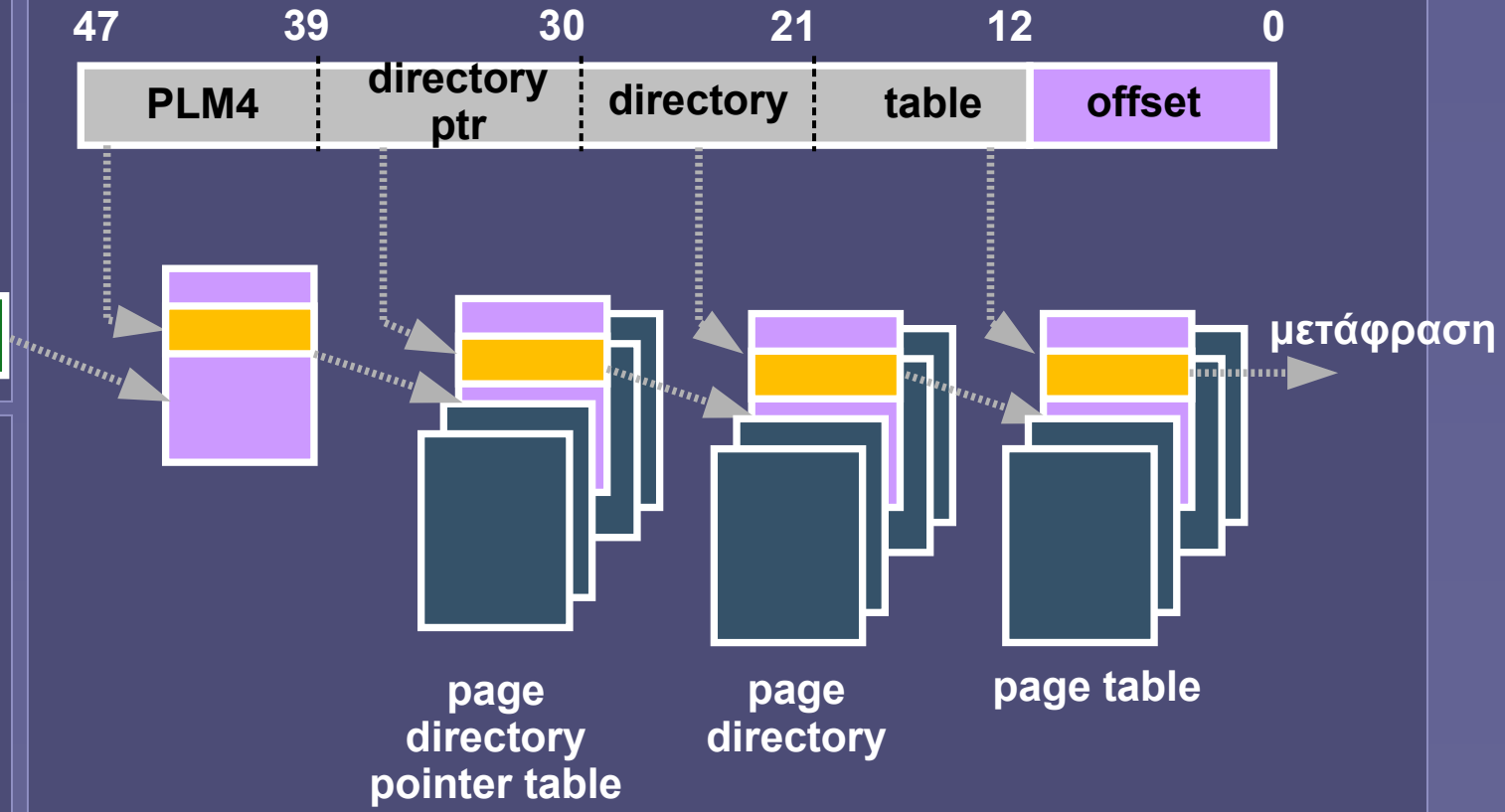
- Τα 64-bit προγράμματα μπορούν να «δουν» έναν πολύ μεγάλο χώρο λογικών διευθύνσεων
  - Π.χ. 0...0x7FFFFFFFFFFF = 128TB
  - Δεν χρησιμοποιείται όλος ο χώρος διευθύνσεων



Πίνακας σελίδων πολλαπλών επιπέδων

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

- Παράδειγμα: αρχιτεκτονική x86-64, σελίδα 4KB
  - 4 (ή και 5) επίπεδα πινάκων για 48-bit (ή 57-bit) διευθύνσεις



Με την οργάνωση πολλαπλών επιπέδων δεν είναι υποχρεωτικό να υπάρχει πληροφορία για όλες τις σελίδες, παρά μόνο όσες χρησιμοποιούνται

Translation-Lookaside Buffer

- Το πρόβλημα με τους πίνακες σελίδων
  - Βρίσκονται στην κύρια μνήμη
  - Για κάθε προσπέλαση μνήμης απαιτείται μια δεύτερη (τουλάχιστον) στον πίνακα σελίδων, επίσης στη μνήμη
    - Μη αποδεκτή χρονική επιβάρυνση
- Translation-Lookaside Buffer (TLB)
  - Μικρή «κρυφή μνήμη» για πρόσφατες μεταφράσεις λογικών διευθύνσεων (στη μονάδα MMU)
    - Ο αριθμός σελίδας χρησιμοποιείται ως ετικέτα (tag)
    - Περιέχει μόνο έγκυρες σελίδες (όχι «απούσες»)
    - Διατηρείται συγχρονισμένο με πίνακες σελίδων
    - Κατά την εναλλαγή διεργασιών πρέπει να «αδειάζει»
      - Εναλλακτικά, χρήση ID διεργασίας μαζί με ετικέτα

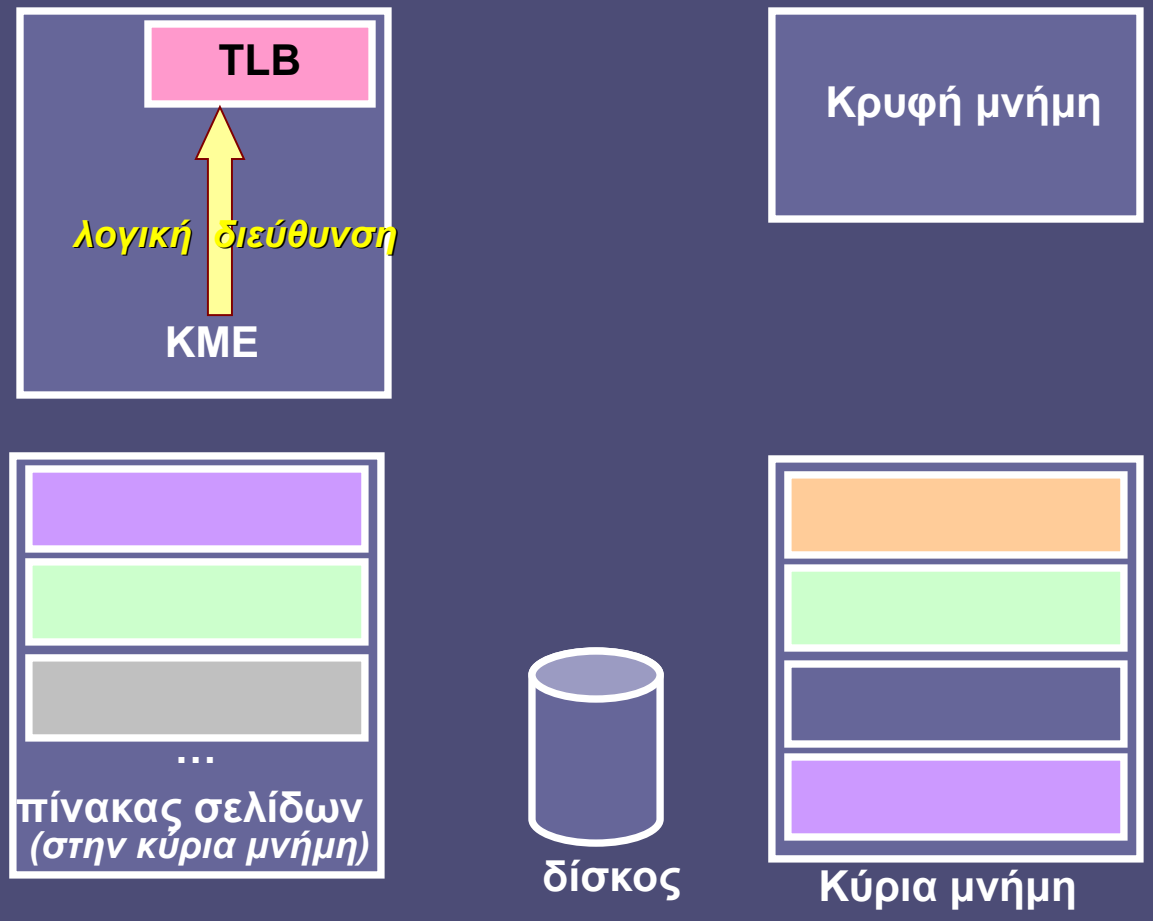
Translation-Lookaside Buffer (2)

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

- TLB miss
  - Προσπέλαση πίνακα σελίδων στη μνήμη και απόκτηση μετάφρασης που λείπει
    - Ενημέρωση από ΛΣ ή από την ίδια τη μονάδα MMU (“page table walking”)
- Χαρακτηριστικά TLB
  - Ξεχωριστά TLB για εντολές και δεδομένα
  - 16-512 θέσεις, 1-2 γραμμές του πίνακα σελίδων ανά θέση
    - Μικρό μέγεθος, τοπικότητα σελίδων πολύ μεγάλη
    - Συχνά πλήρως προσεταιριστικό
  - Προσπέλαση < 1 κύκλο ρολογιού
  - Παρατηρούμενο Miss rate: 0.01% - 1%
  - Όπως και στις πραγματικές κρυφές μνήμες, μπορεί να υπάρχει L1 και L2 TLB

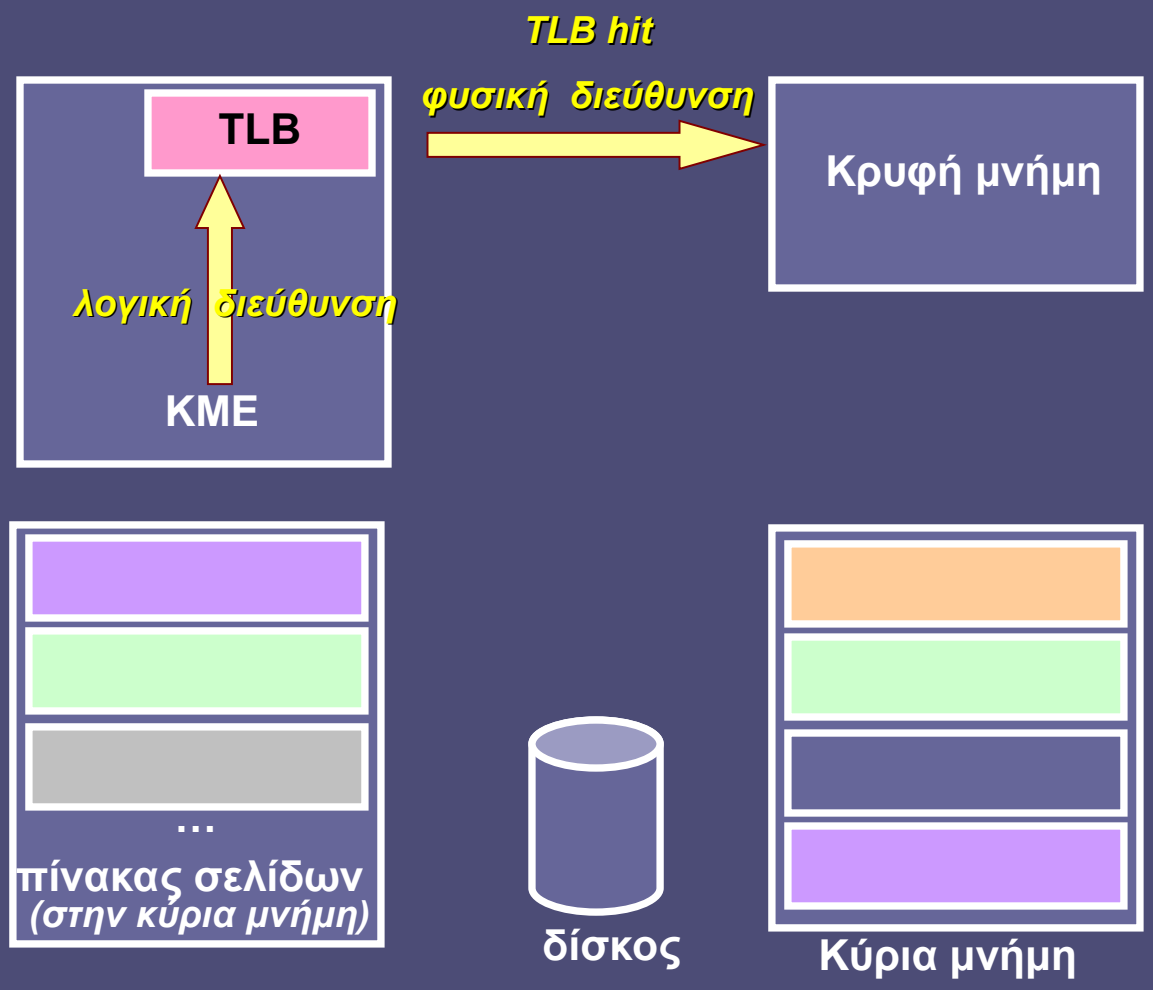
Προσπέλαση μνήμης: η συνολική εικόνα

- Ιεραρχία Μνήμης
- Εικονική Μνήμη



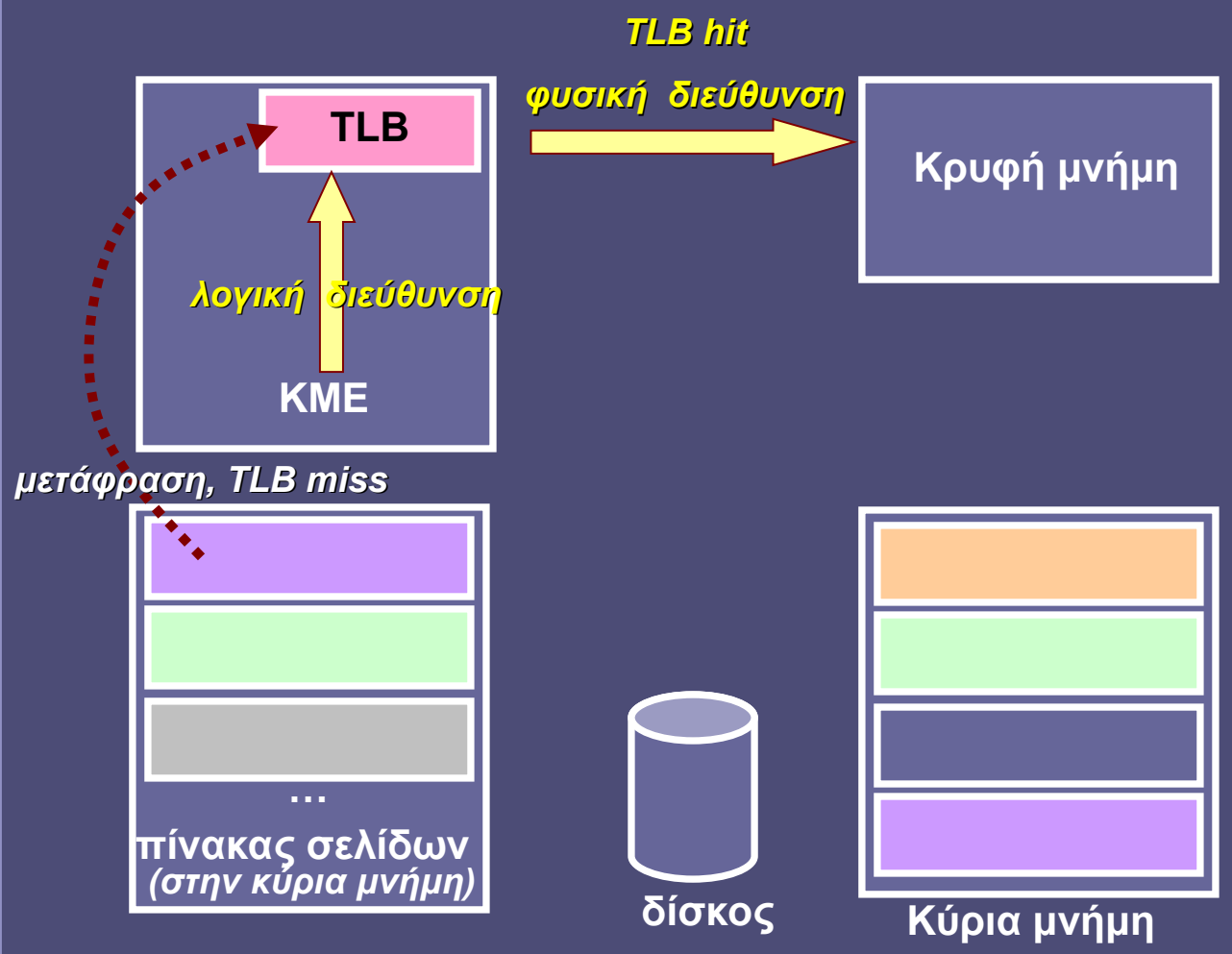
Προσπέλαση μνήμης: η συνολική εικόνα

- Ιεραρχία Μνήμης
- Εικονική Μνήμη



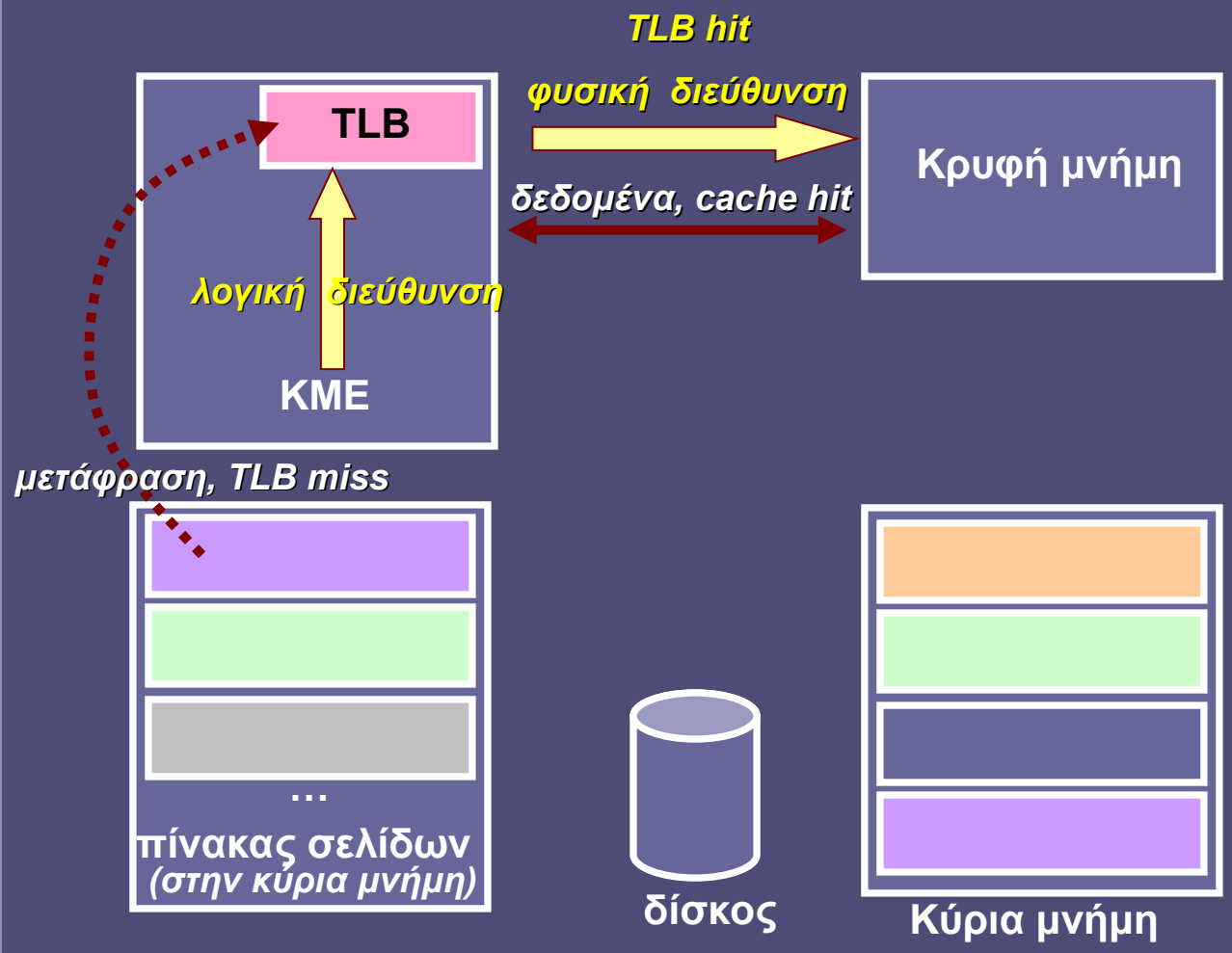
Προσπέλαση μνήμης: η συνολική εικόνα

- Ιεραρχία Μνήμης
- Εικονική Μνήμη



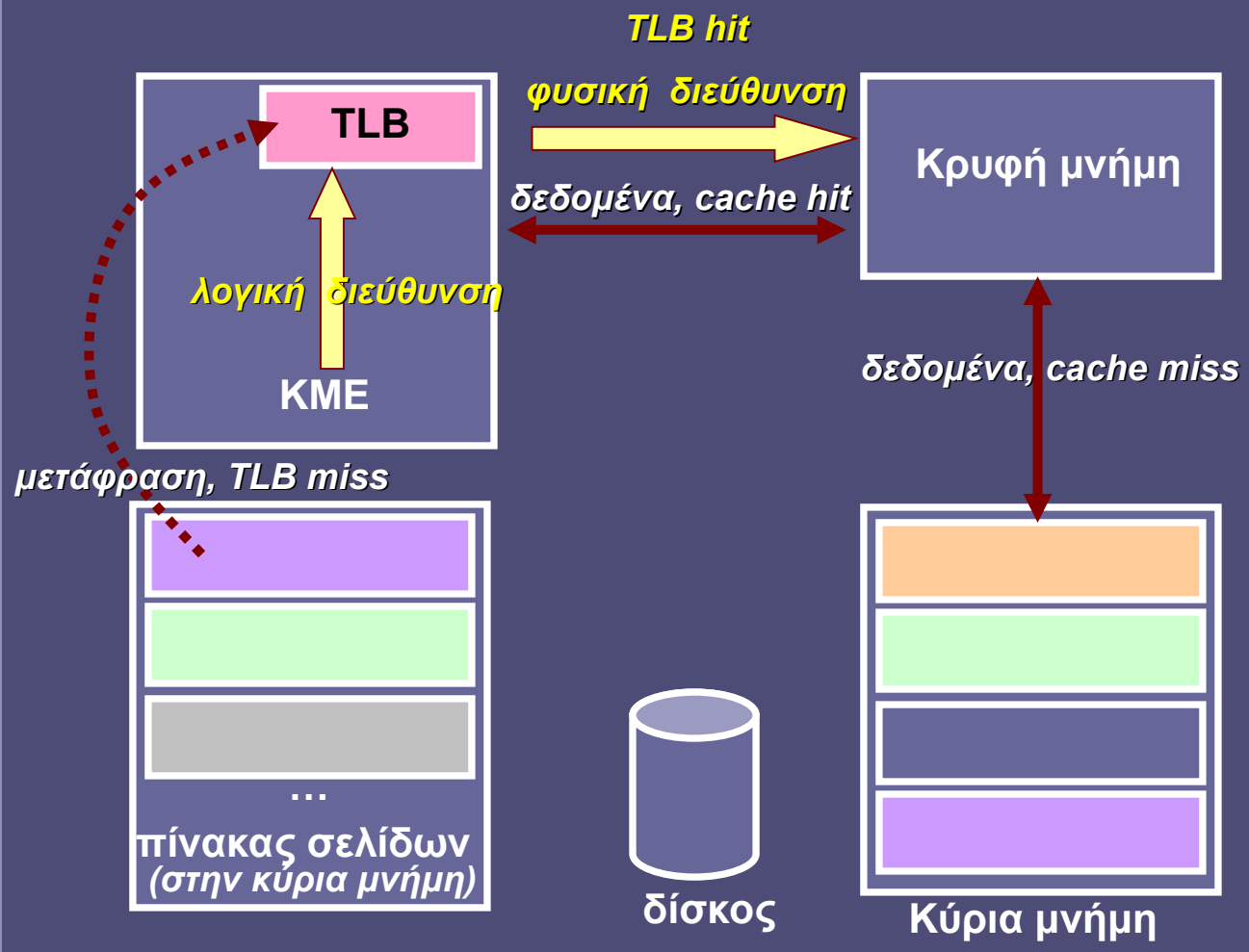
Προσπέλαση μνήμης: η συνολική εικόνα

- Ιεραρχία Μνήμης
- Εικονική Μνήμη



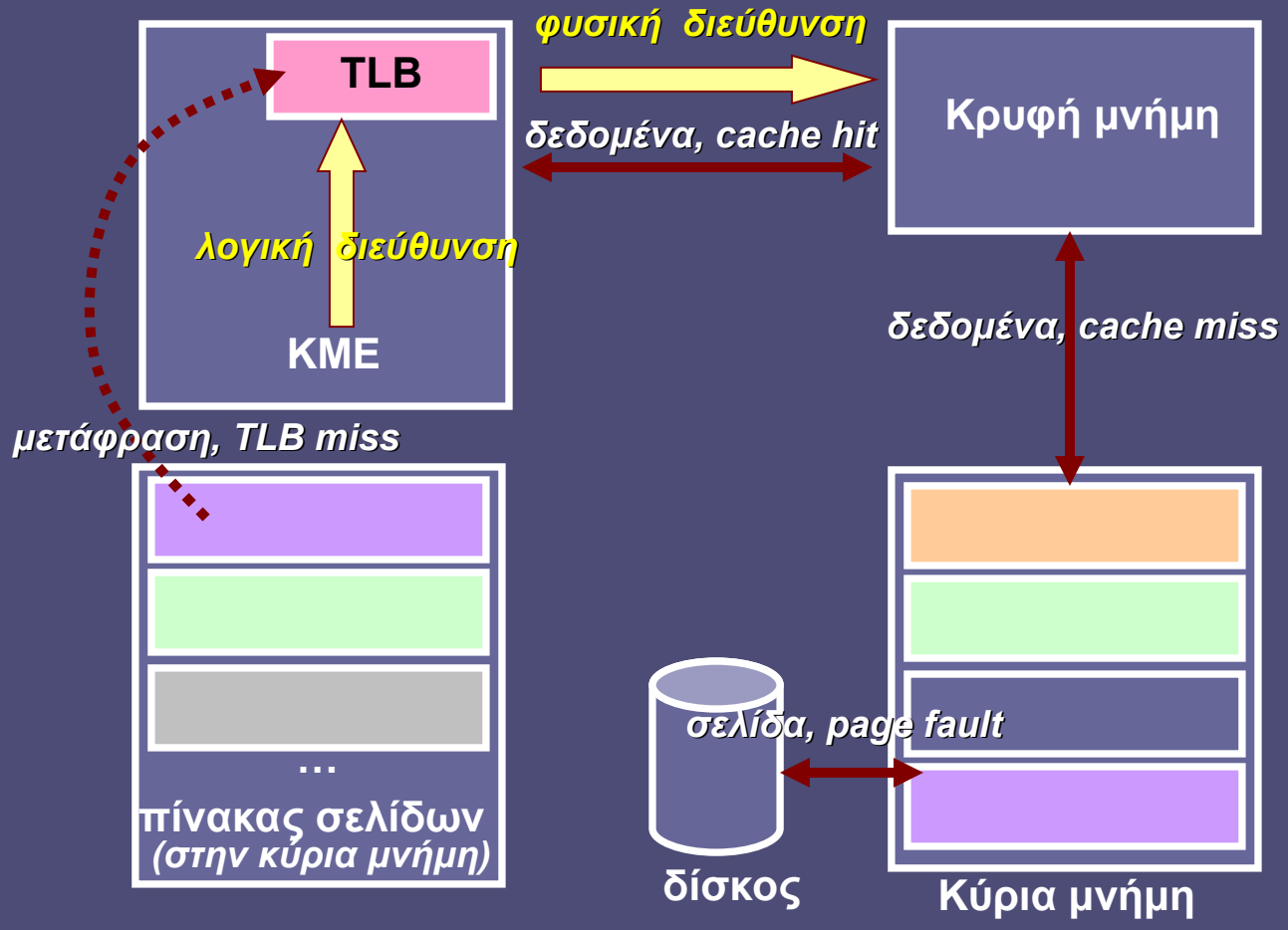
Προσπέλαση μνήμης: η συνολική εικόνα

- Ιεραρχία Μνήμης
- Εικονική Μνήμη



Προσπέλαση μνήμης: η συνολική εικόνα

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

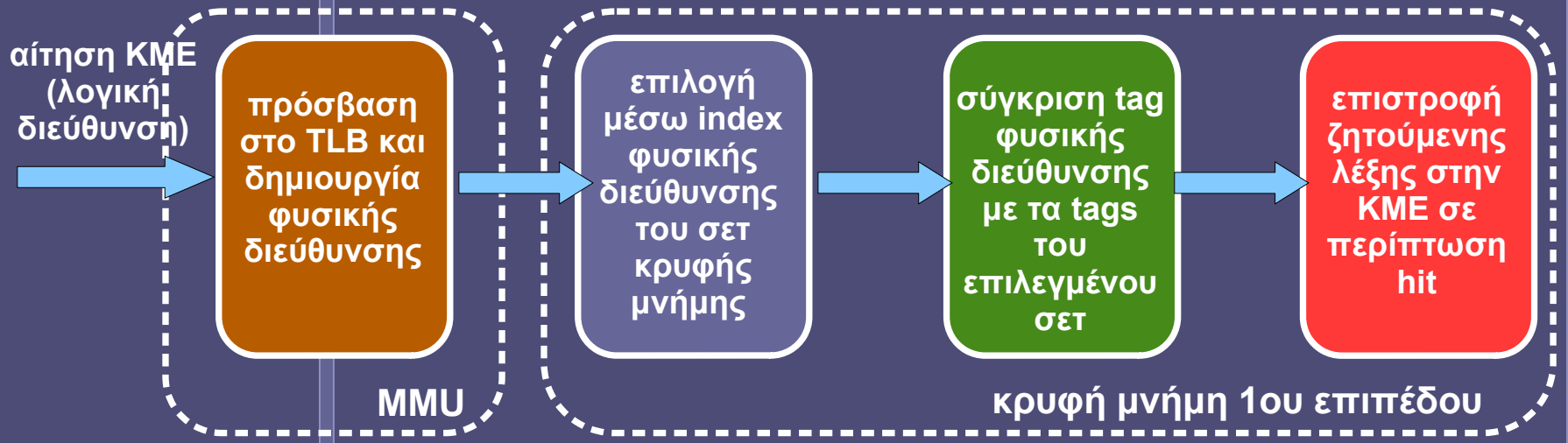


TLB και κρυφή μνήμη 1ου επιπέδου

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η κρυφή μνήμη 1ου επιπέδου (κοντά στην ΚΜΕ) πρέπει να είναι πολύ γρήγορη σε περίπτωση hit

- Με την εισαγωγή της εικονικής μνήμης παρεμβάλλεται και το στάδιο της μετάφρασης από λογική σε φυσική διεύθυνση
- Πώς μπορούμε να επιταχύνουμε τη διαδικασία;



Virtually Indexed Physically Tagged (VIPT) caches

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Με σελίδες 4KB (12 bits offset) ο αριθμός σετ της κρυφής μνήμης VIPT είναι περιορισμένος – OK για 1ο επίπεδο

Αν εξασφαλίσουμε ότι το index είναι μέσα στο offset της λογικής διεύθυνσης, δεν χρειάζεται μετάφραση για να το χρησιμοποιήσουμε

- Το offset παραμένει ίδιο στη φυσική διεύθυνση