

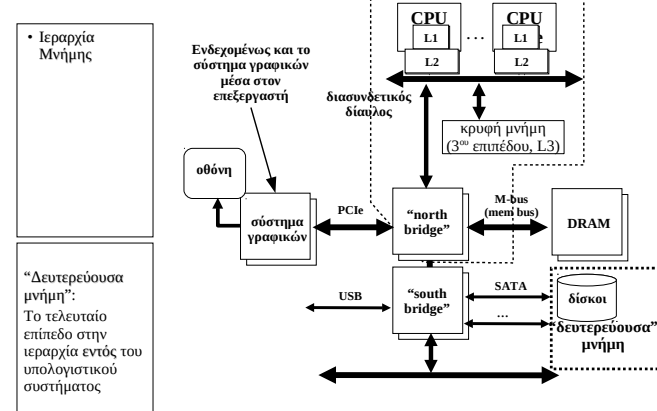
## Εικονική Μνήμη (και ο ρόλος της στην ιεραρχία μνήμης)

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



Μ.Στεφανιδάκης

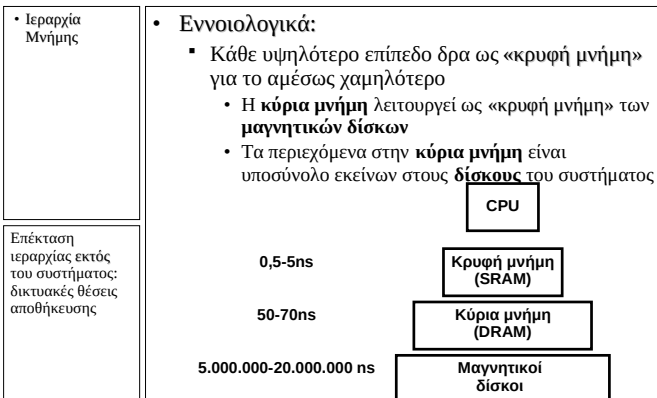
## Ιεραρχία Μνήμης



Αρχιτεκτονική Υπολογιστών – “Εικονική Μνήμη”

2

## Επεκτείνοντας την Ιεραρχία Μνήμης



Αρχιτεκτονική Υπολογιστών – “Εικονική Μνήμη”

3

## 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
      - Πιθανότητα αντικαθιστώντας άλλα τμήματα δεδομένων
      - Τα τελευταία μεταφέρονται πίσω στους δίσκους

Αρχιτεκτονική Υπολογιστών – “Εικονική Μνήμη”

4

## Πριν την εικονική μνήμη: overlays

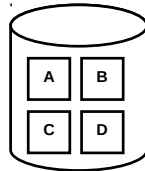
- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η τεχνική των overlays απαιτούσε από τον προγραμματιστή να χειρίζεται τις λεπτομέρειες (π.χ. διευθύνσεις!) φόρτωσης και κλήσης των υποπρογραμμάτων!

### Κύρια μνήμη

```
main() {  
  swap-in(D)  
  call D1  
  
  swap-in(B)  
  Call B2  
}
```

### δίσκος



- Καταλληλότερο για στατικά δεδομένα
  - υποπρογράμματα

## Πριν την εικονική μνήμη: overlays

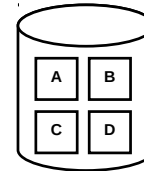
- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η τεχνική των overlays απαιτούσε από τον προγραμματιστή να χειρίζεται τις λεπτομέρειες (π.χ. διευθύνσεις!) φόρτωσης και κλήσης των υποπρογραμμάτων!

### Κύρια μνήμη

```
main() {  
  swap-in(D) ←  
  call D1  
  
  swap-in(B)  
  Call B2  
}
```

### δίσκος



- Καταλληλότερο για στατικά δεδομένα
  - υποπρογράμματα

## Πριν την εικονική μνήμη: overlays

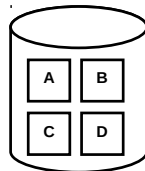
- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η τεχνική των overlays απαιτούσε από τον προγραμματιστή να χειρίζεται τις λεπτομέρειες (π.χ. διευθύνσεις!) φόρτωσης και κλήσης των υποπρογραμμάτων!

### Κύρια μνήμη

```
main() {  
  swap-in(D) ←  
  call D1  
  
  swap-in(B) ←  
  Call B2  
}
```

### δίσκος



- Καταλληλότερο για στατικά δεδομένα
  - υποπρογράμματα

## 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;

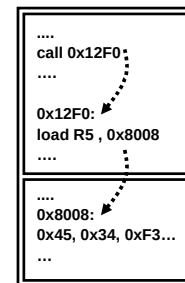
### Address Space

- Εκτελούμενο πρόγραμμα στη μνήμη:
- Διευθύνσεις κώδικα
  - Εντολές διακλάδωσης
- Διευθύνσεις δεδομένων
  - Εντολές load-store

### Πριν την εικονική μνήμη:

- Φυσικές διευθύνσεις
- Άμεση αντιστοιχία με διευθύνσεις κύριας μνήμης

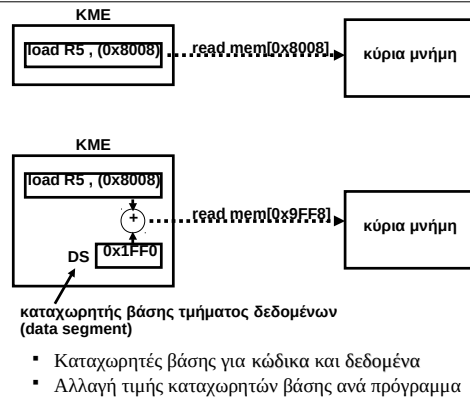
### Κύρια μνήμη



## Πολυπρογραμματισμός πριν την εικονική μνήμη

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Χρήση καταχωρητών βάσης (χωρίς άλλη υποστήριξη) σε χαμηλού κόστους επεξεργαστές (π.χ. Intel 8086).  
Διευκόλυνση τοποθέτησης προγραμμάτων οπουδήποτε στη μνήμη.



## Μεταβαίνοντας σε εικονικές διευθύνσεις

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

### • Το προηγούμενο σχήμα

- Εισήγαγε την αποσύνδεση των λογικών διευθύνσεων των προγραμμάτων από τις φυσικές διευθύνσεις κύριας μνήμης

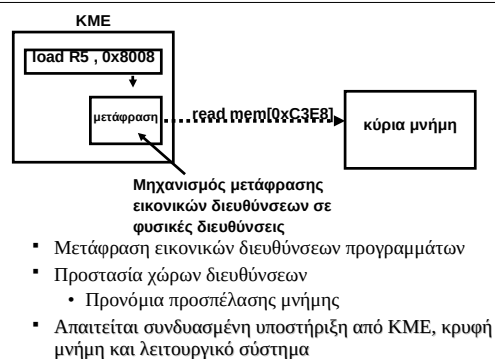
- Με απλή αντιστοιχία:  
φυσική διεύθυνση = λογική διεύθυνση + καταχωρητής βάσης

- Απαιτείται υποστήριξη από το υλικό (KME)
- 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 όμως!

## Εικονικές Διευθύνσεις

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Ο κύριος ρόλος της εικονικής μνήμης στα σημερινά υπολογιστικά συστήματα, με την άφθονη κύρια μνήμη, είναι η υποστήριξη και προστασία των εικονικών χώρων διευθύνσεων ανά πρόγραμμα



## Η εικονική μνήμη (ξανά)

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

### • Ποιος ο ρόλος της;

1. Η χρήση εικονικής μνήμης, πέρα από όση είναι πραγματικά διαθέσιμη
2. Η αντιστοίχιση εικονικών διευθύνσεων σε φυσικές και η διαχείριση των προνομιών προσπέλασης
  - Σημαντικότερο σήμερα!

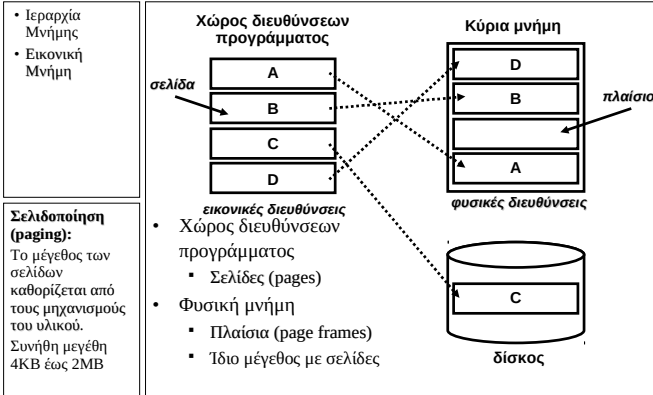
### • Ποιος τη διαχειρίζεται;

- Διαχείριση από το λειτουργικό σύστημα
- Υποστήριξη από το υλικό (KME/κρυφή μνήμη)

### • Πώς υλοποιείται;

- Μετακίνηση τμημάτων μνήμης από/προς τους δίσκους
- Εκμετάλλευση αρχής τοπικότητας
  - Μερικά μέρη μόνο των προγραμμάτων είναι «ενεργά» κάθε στιγμή

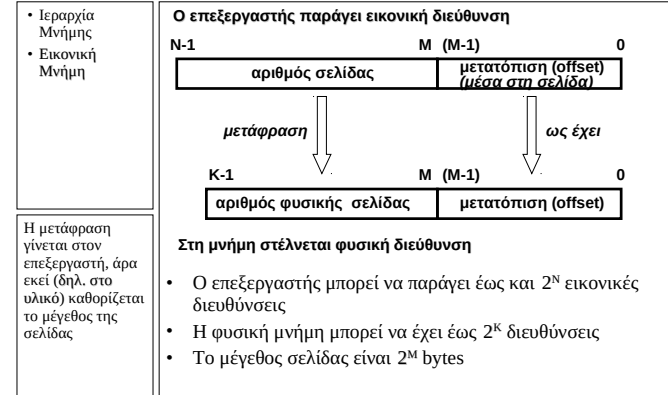
## Βασικό σχήμα Εικονικής Μνήμης



Αρχιτεκτονική Υπολογιστών – “Εικονική Μνήμη”

13

## Σελίδες και εικονικές διευθύνσεις



Αρχιτεκτονική Υπολογιστών – “Εικονική Μνήμη”

14

## Χαρακτηριστικά Σελίδων

- Ιεραρχία Μνήμης
  - Εικονική Μνήμη

- Σταθερό μέγεθος (4KB-2MB)
  - Για την προσπέλαση οποιασδήποτε θέσης μνήμης της σελίδας απαιτείται μία λέξη εικονικής διεύθυνσης (page+offset)
  - Ευκολία τοποθέτησης και αντικατάστασης σελίδων στην κύρια μνήμη
  - Βέλτιστο μέγεθος για μεταφορά από-προς δίσκο
  - Ενδεχομένως ένα μέρος της σελίδας μένει αχρησιμοποίητο

Αρχιτεκτονική Υπολογιστών – “Εικονική Μνήμη”

15

## Σελιδοποίηση κατ'απαίτηση

- 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 προσπελαστεί 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”
  - Page faults: μεγάλο κόστος σε κύκλους αναμονής
    - 1-10Μκύκλοι

Αρχιτεκτονική Υπολογιστών – “Εικονική Μνήμη”

16

## Κρίσιμα σημεία στη σχεδίαση εικονικής μνήμης

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

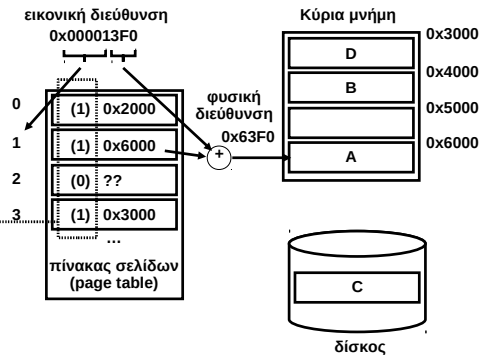
Ακόμα και μικρή μείωση στην εμφάνιση page faults μπορεί να έχει σημαντικό όφελος για την απόδοση του συστήματος

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

## Μετάφραση εικονικών διευθύνσεων

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Presence bit: .....  
βρίσκεται η σελίδα στην κύρια μνήμη;



## Πίνακας σελίδων

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

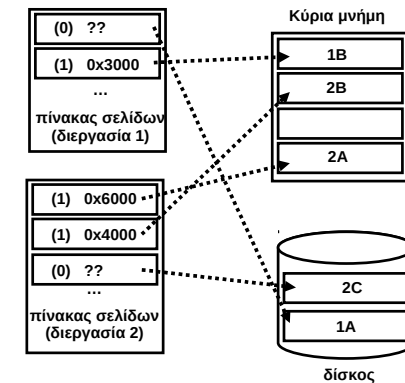
Τι συμβαίνει στην περίπτωση πολλαπλών προγραμμάτων;

- Πού βρίσκεται;
  - Στην κύρια μνήμη!
- Πόσες θέσεις διαθέτει;
  - Θεωρητικά: ίσες με τον μέγιστο αριθμό σελίδων
    - Για  $2^{20}$  σελίδες με 4 bytes ανά γραμμή, απαιτούνται 4MB
  - Πρακτικά: πίνακες πολλαπλών επιπέδων
  - Πιθανόν: σελιδοποίηση πινάκων (!)
    - Αρκεί να υπάρχει πάντα στη μνήμη ένα μέρος του πίνακα
- Ποια πρόσθετη πληροφορία χρειάζεται;
  - Βρίσκεται η σελίδα στη μνήμη; Έχει αλλάξει;
  - Αν όχι, σε ποιο σημείο του δίσκου είναι;
    - Συχνά η πληροφορία αυτή φυλάσσεται μέσα στον πίνακα σελίδων
  - Ποια σελίδα βρίσκεται σε κάθε πλαίσιο μνήμης;
  - Πόσο πρόσφατα χρησιμοποιήθηκε;
    - Δομές του ΛΣ για αλγόριθμο αντικατάστασης σελίδων

## Πίνακες σελίδων και πολλαπλά προγράμματα

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Κάθε διεργασία έχει τους δικούς της πίνακες σελίδων. Κατά την εναλλαγή διεργασιών αλλάζει και ο καταχωρητής-δείκτης στους πίνακες



## Translation-Lookaside Buffer

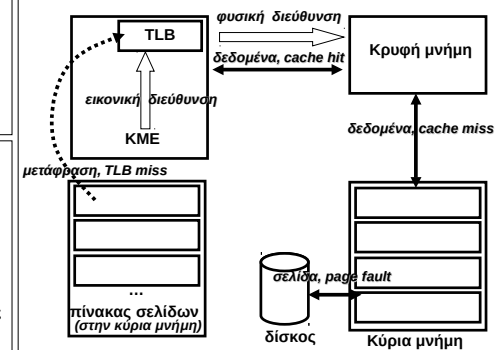
- Ιεραρχία Μνήμης
- Εικονική Μνήμη

- Το πρόβλημα με τους πίνακες σελίδων
  - Βρίσκονται στην κύρια μνήμη
  - Για κάθε προσπέλαση μνήμης απαιτείται μια δεύτερη(!)
    - Για τον πίνακα σελίδων
    - Μη αποδεκτή χρονική επιβάρυνση
- Translation-Lookaside Buffer (TLB)
  - Μικρή «κρυφή μνήμη» για πρόσφατες μεταφράσεις εικονικών διευθύνσεων
    - Μέσα στον επεξεργαστή
    - 16-512 θέσεις, 1-2 γραμμές του πίνακα σελίδων ανά θέση
    - Προσπέλαση < 1 κύκλο ρολογιού
    - Παρατηρούμενο Miss rate: 0.01% - 1%
    - Σε περίπτωση miss: ενημέρωση από ΛΣ ή από την ίδια την ΚΜΕ (μηχανισμός page table walking)

## Προσπέλαση μνήμης: η συνολική εικόνα

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Στο σχήμα η κρυφή μνήμη δέχεται φυσικές διευθύνσεις. Υπάρχουν αρχιτεκτονικές με εικονική κρυφή μνήμη ή συνδυασμό εικονικής-φυσικής κρυφής μνήμης



## Εικονική Μνήμη και Προστασία Προσπέλασης

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

- Προστασία προσπέλασης σελίδων
  - Με διαφορετικούς πίνακες σελίδων ανά διεργασία είναι αδύνατη η προσπέλαση «ξένων» σελίδων
  - Δικαιώματα προσπέλασης ανά σελίδα
  - User mode και Supervisor Mode
    - Σε user mode δεν είναι δυνατή η προσπέλαση του TLB, του πίνακα σελίδων και των αντίστοιχων καταχωρητών συστήματος
    - Υπάρχουν αρχιτεκτονικές με περισσότερα από 2 επίπεδα προνομίων
  - Ελεγχόμενη προσπέλαση συναρτήσεων ΛΣ
    - Call gates: ελεγχόμενη εκτέλεση συναρτήσεων ΛΣ από χρήστη, στο επίπεδο όμως των προνομίων του χρήστη → δεν είναι δυνατή η προσπέλαση «ξένων» δεδομένων!