Περιεχόμενα μαθήματος

- Εισαγωγή
- Διεργασίες & Νήματα
- Χρονοδρομολόγηση
- Συγχρονισμός
- Διαχείριση μνήμης και σύστημα εικονικής μνήμης
- Συστήματα αρχείων
- Χρονοδρομολόγηση Ε/Ε

Λειτουργικό Σύστημα

Ορισμός:

Το Λειτουργικό Σύστημα (ΛΣ) είναι ένα πρόγραμμα-διεπαφή ανάμεσα στο υλικό ενός υπολογιστικού συστήματος και τους χρήστες του (εφαρμογές).

Επιτελεί ρόλους όπως:

- Ελεγχόμενη πρόσβαση στους πόρους του συστήματος (**ασφάλεια**)
- Διαμοιρασμός των πόρων του συστήματος (**επίδοση** και **δικαιοσύνη**)
- Υπηρεσίες για **εύκολη** πρόσβαση στο υλικό
  - Συστήματα αρχείων
  - Δικτυακά πρωτόκολλα
- Κοινή διεπαφή με υλικό για προγραμματιστές (POSIX)

Εναλλακτικοί ορισμοί ΛΣ

top-down

Παρέχει στα προγράμματα εύκολη και αποδοτική πρόσβαση στους πόρους του συστήματος.

bottom-up

"Παρέχει μια συστηματοποιημένη και ελεγχόμενη κατανομή των επεξεργαστών, των μνημών και των άλλων συσκευών Ε/Ε, ανάμεσα στα διάφορα προγράμματα-πελάτες που ανταγωνίζονται μεταξύ τους για να τα χρησιμοποιήσουν" (Tanenbaum, 2001)

Υπολογιστικά Συστήματα

- **Υλικό**

  ΚΜΕ/Μνήμη/Μονάδες Ε/Ε

- **Λειτουργικό Σύστημα**

  Ελέγχει και συντονίζει την κοινόχρηστη χρήση του υλικού (των πόρων του υπολογιστή) μεταξύ πολλών εφαρμογών και χρηστών.

- **Προγράμματα εφαρμογών**

  Προσδιορίζουν τον τρόπο με τον οποίο οι πόροι του συστήματος χρησιμοποιούνται για την επίλυση συγκεκριμένων προβλημάτων. πχ επεξεργαστές κειμένου, compilers, web browsers, συστήματα βάσεων δεδομένων, παιχνίδια ...

- **Χρήστες**

  Άνθρωποι, μηχανές, άλλοι υπολογιστές

Απαιτήσεις ΛΣ

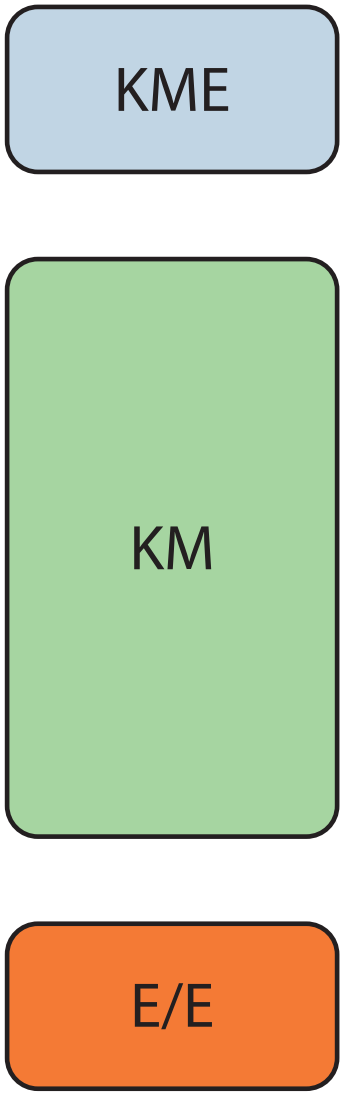
- Προσωπικοί υπολογιστές (PCs):
  - Χαμηλοί χρόνοι απόκρισης (responsiveness)
- Κεντρικά υπολογιστικά συστήματα (mainframes):
  - Υψηλή επίδοση, δίκαιη διαχείριση πόρων
- Υπολογιστές χειρός (palm PCs):
  - Ευκολία στη χρήση, οικονομία στους πόρους, χαμηλή κατανάλωση ενέργειας
- Ενσωματωμένοι υπολογιστές (embedded systems):
  - Απόκριση σε πραγματικό χρόνο (real-time)

Λειτουργικά Συστήματα και Οργάνωση Υπολογιστών

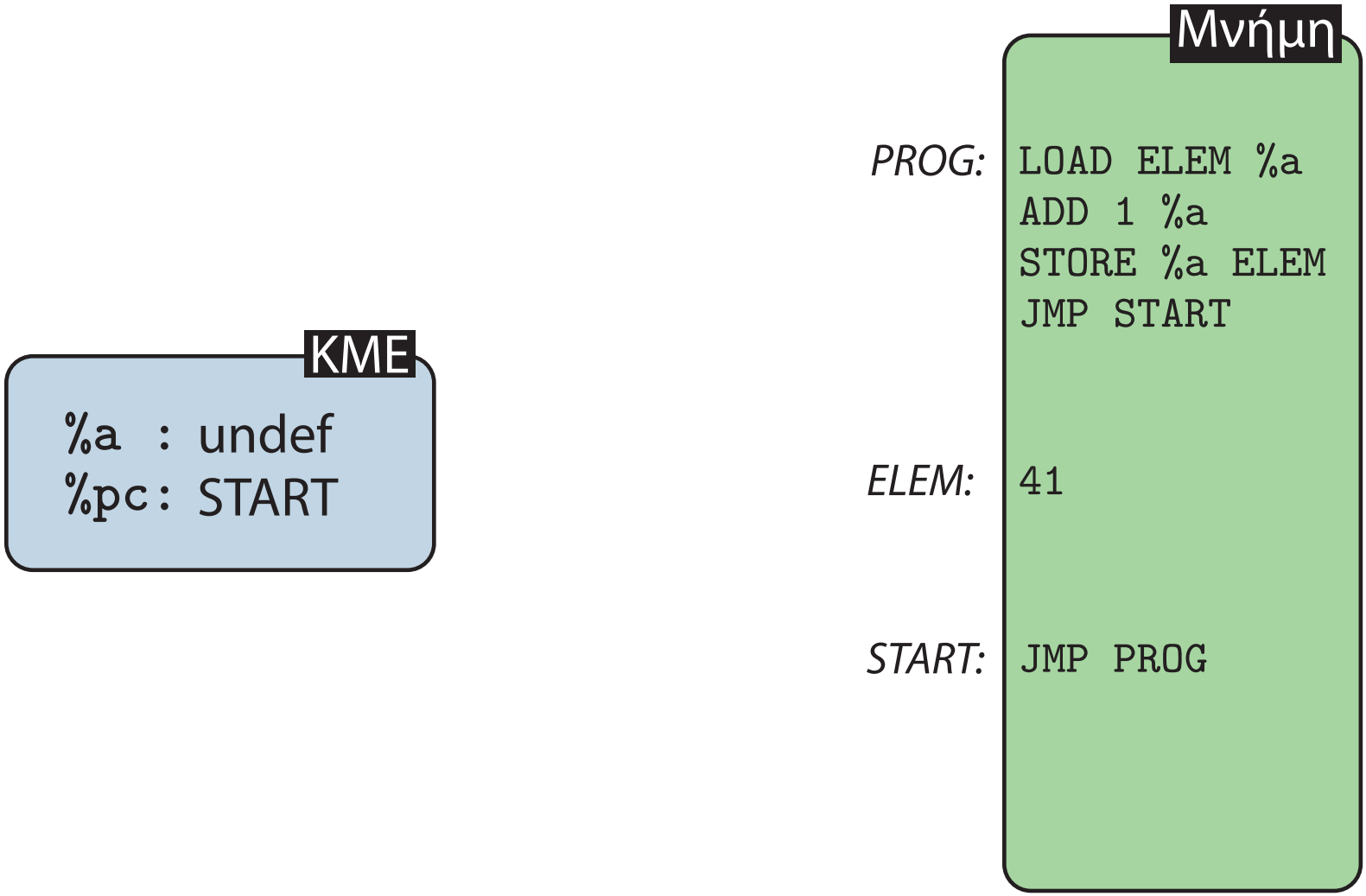
- Σχέση Εξάρτησης μεταξύ ΛΣ και αρχιτεκτονικής
- Πολλές λειτουργίες των ΛΣ χρειάζονται υποστήριξη από το Υλικό
- Νέες δυνατότητες του Υλικού χρειάζονται υποστήριξη από το ΛΣ, ώστε να είναι διαθέσιμες στις εφαρμογές

Οργάνωση Υπολογιστών (Von Neumann)

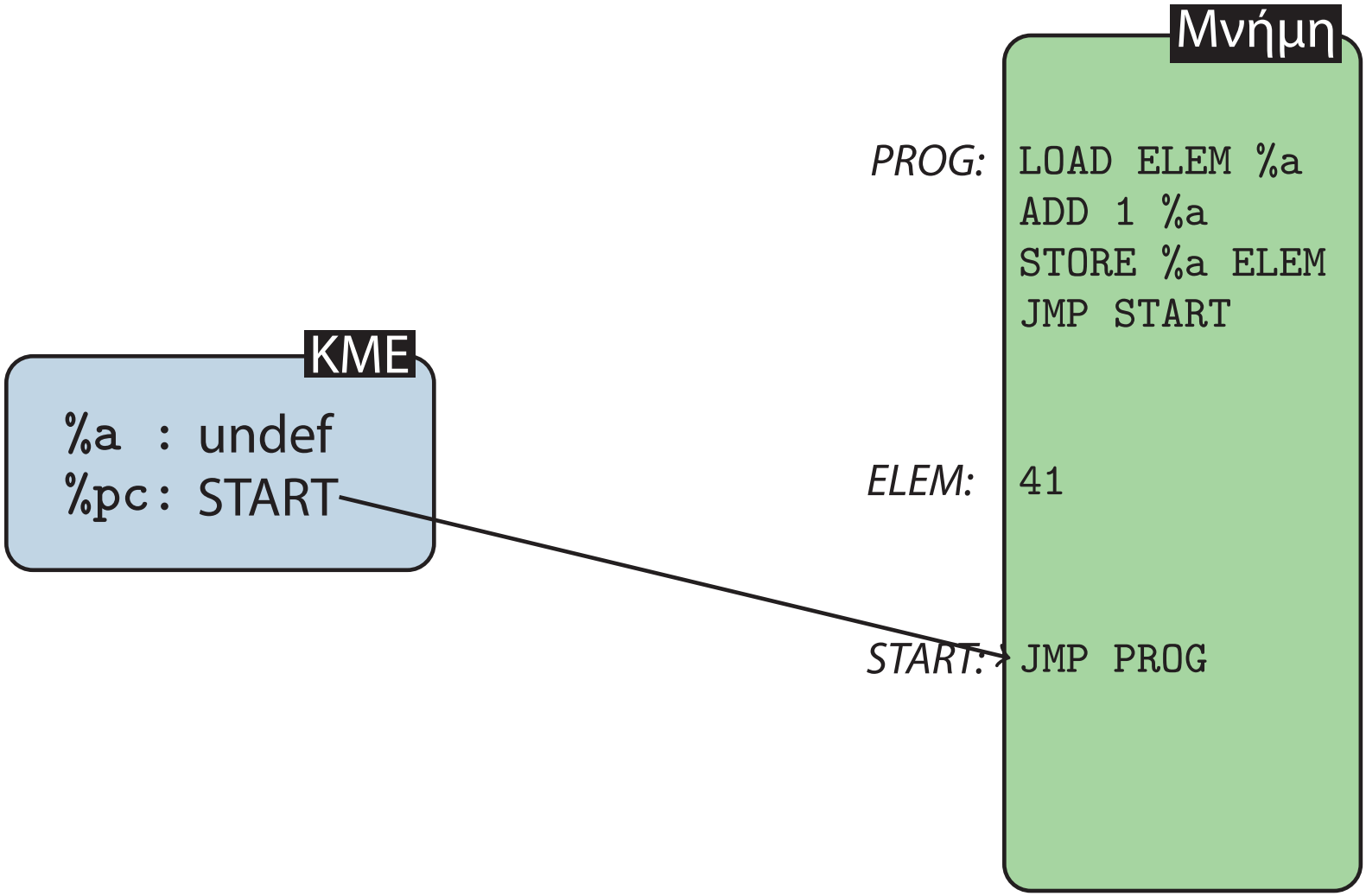
- Κύρια Μνήμη -- ΚΜ (MEM)
  - Γραμμικός χώρος αποθήκευσης
  - Δεδομένα & Πρόγραμμα στην ΚΜ!
- Κεντρική Μονάδα Επεξεργασίας -- ΚΜΕ (CPU)
  - Καταχωρητές
  - Εντολές (π.χ. `ADD`/`LOAD`/`STORE`)
- Είσοδος/Έξοδος -- Ε/Ε (I/O)
  - Διακοπές (Interrupts)



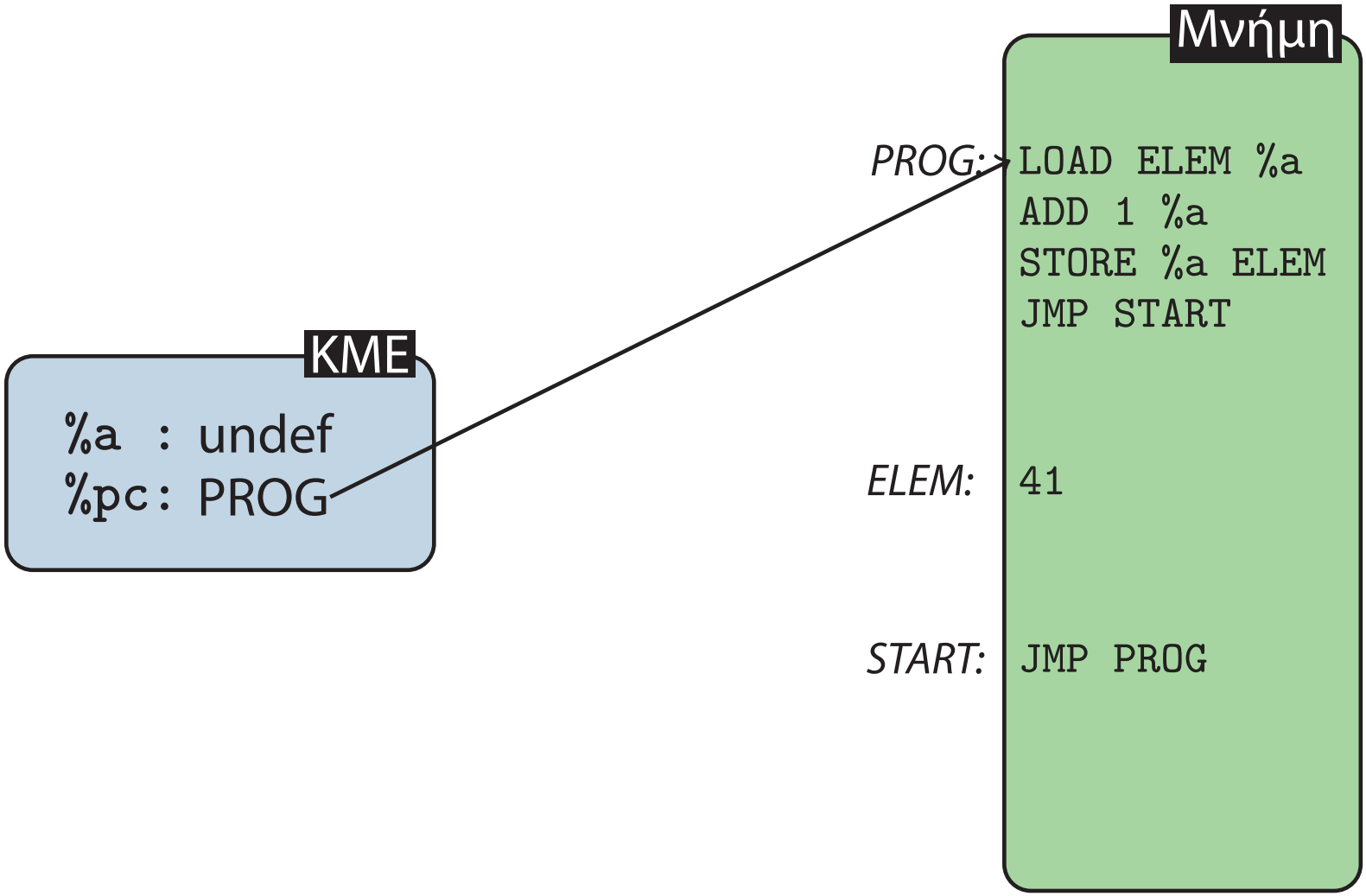
Μοντέλο Εκτέλεσης Προγραμμάτων



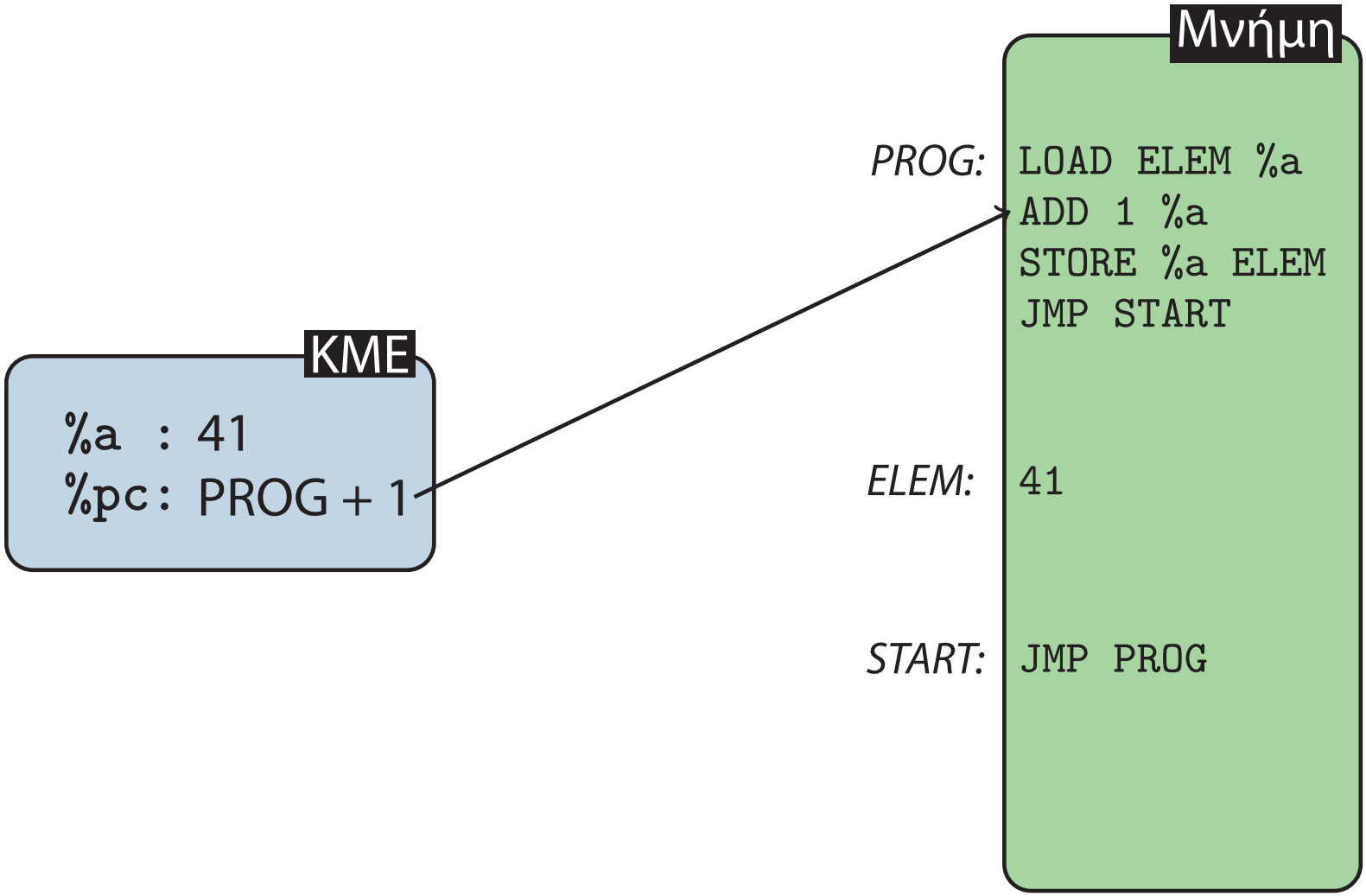
Μοντέλο Εκτέλεσης Προγραμμάτων



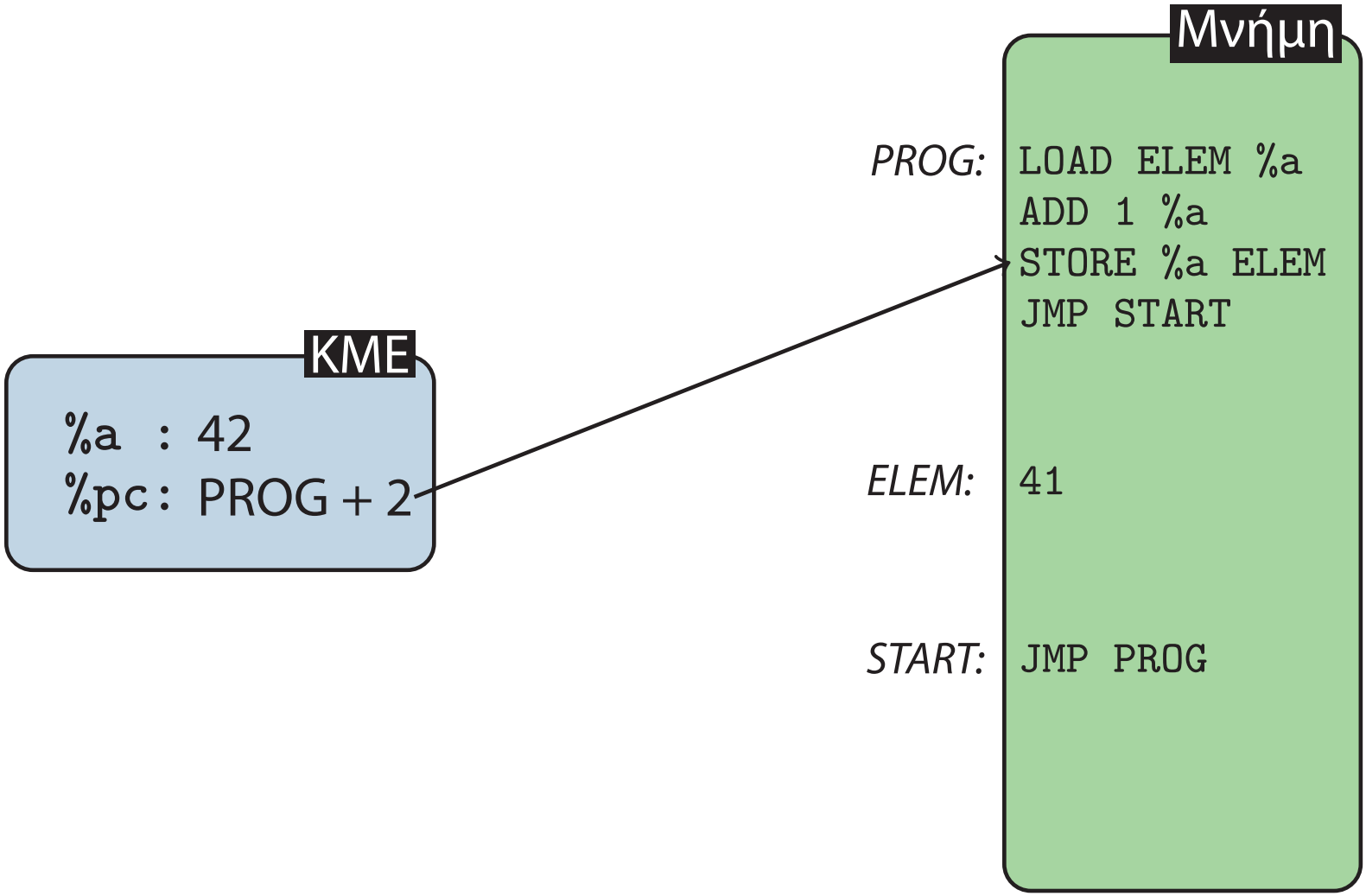
Μοντέλο Εκτέλεσης Προγραμμάτων



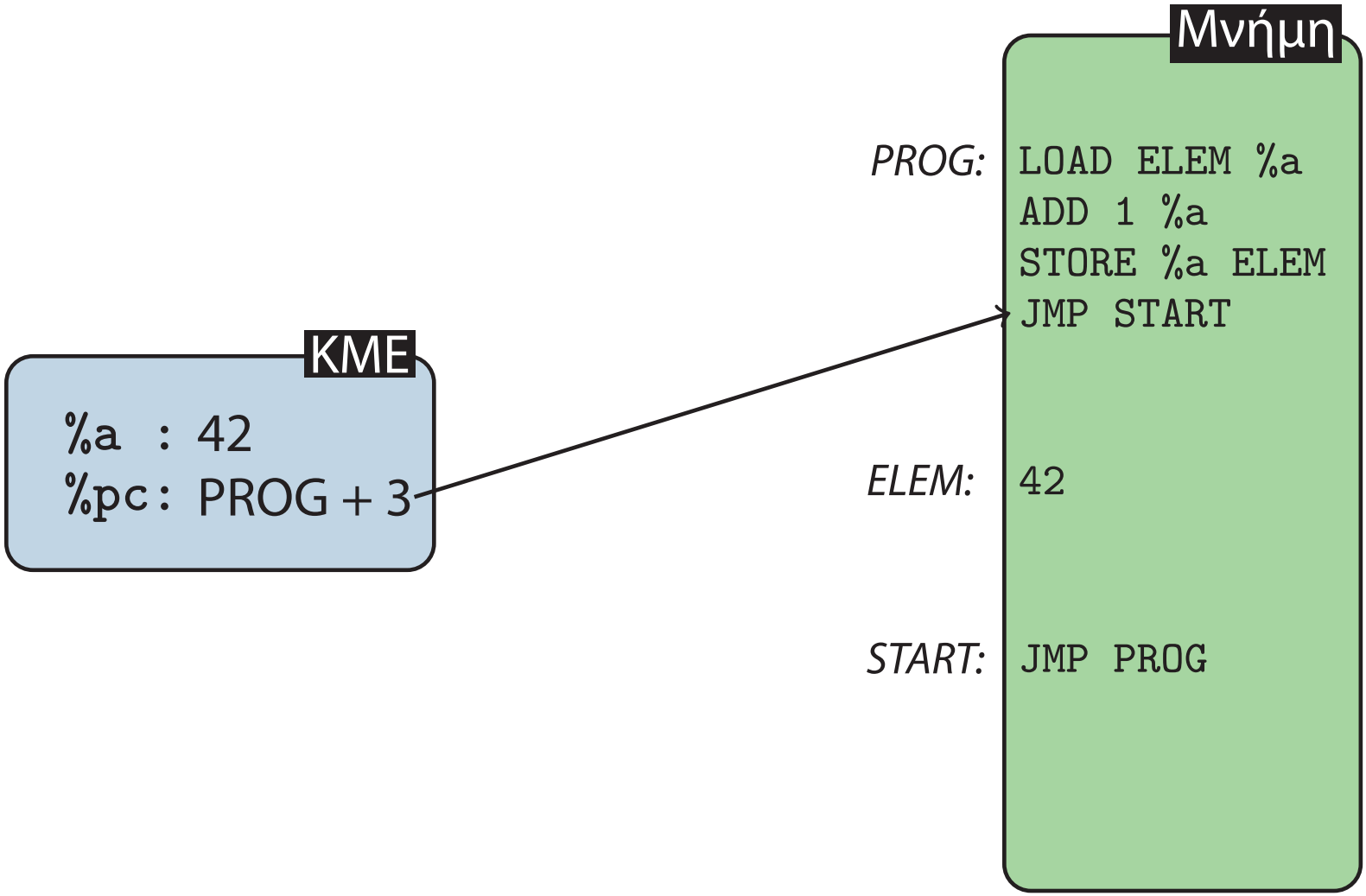
Μοντέλο Εκτέλεσης Προγραμμάτων



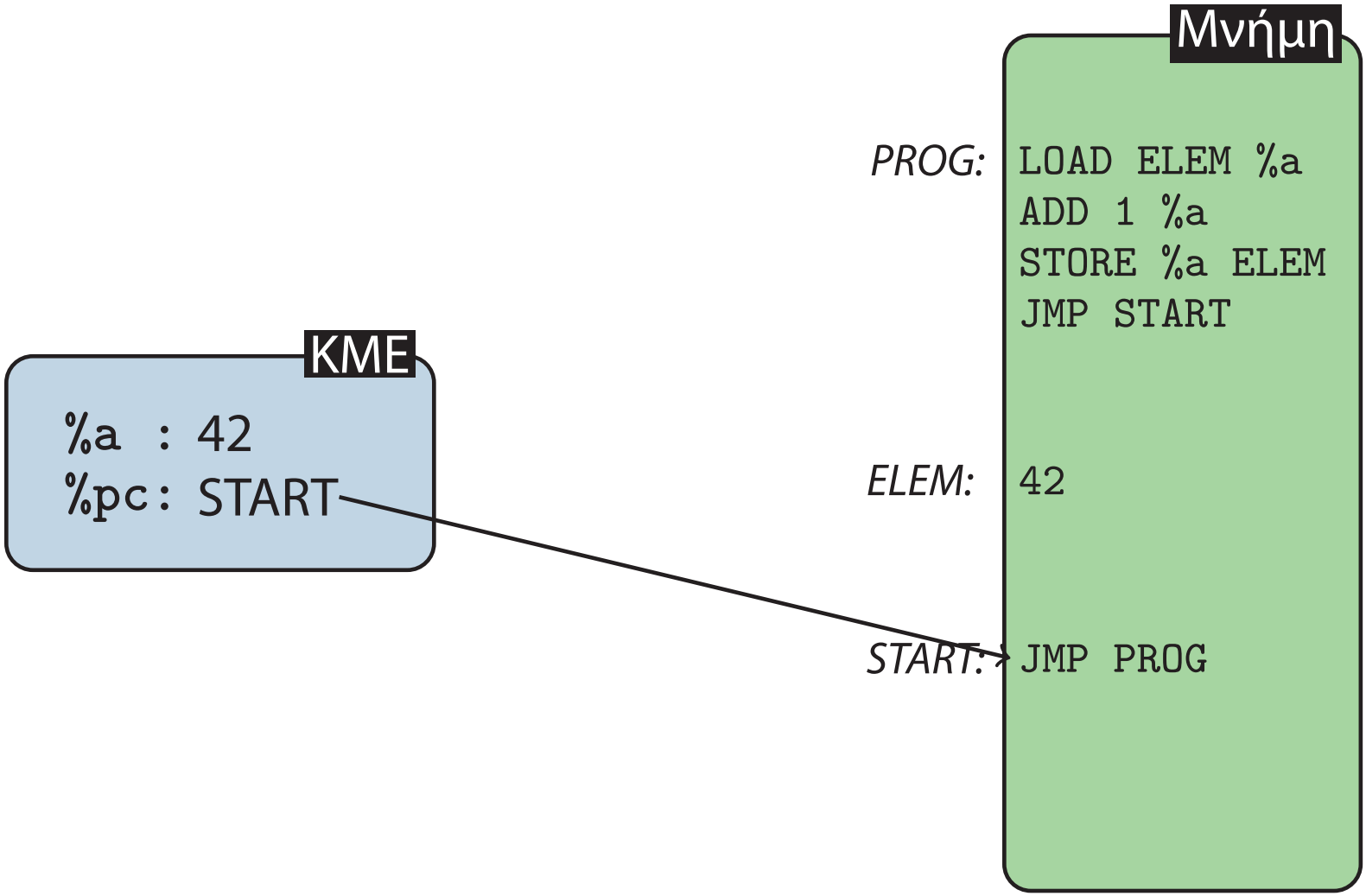
Μοντέλο Εκτέλεσης Προγραμμάτων



Μοντέλο Εκτέλεσης Προγραμμάτων



Μοντέλο Εκτέλεσης Προγραμμάτων



Προβλήματα (προηγ.) Μοντέλου Εκτέλεσης

?

Προβλήματα (προηγ.) Μοντέλου Εκτέλεσης

Εκτέλεση πολλαπλών προγραμμάτων

Εκτέλεση πολλαπλών προγραμμάτων

Διαμοιρασμός χρόνου

Τα πολλαπλά προγράμματα προς εκτέλεση μοιράζονται τον επεξεργαστή (ισχύει για διάφορους πόρους συστήματος).

Εκτέλεση πολλαπλών προγραμμάτων

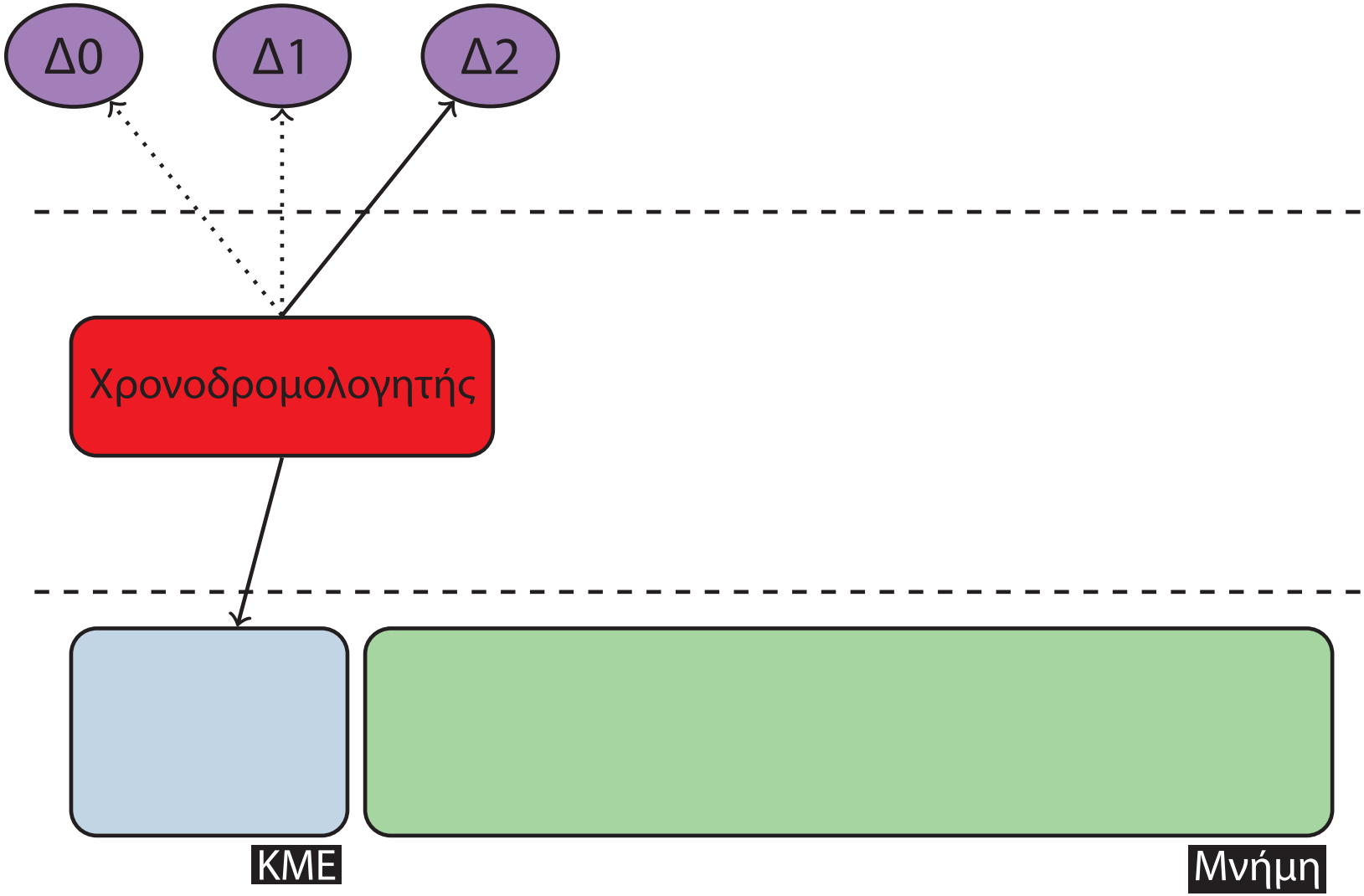
Διαμοιρασμός χρόνου

Τα πολλαπλά προγράμματα προς εκτέλεση μοιράζονται τον επεξεργαστή (ισχύει για διάφορους πόρους συστήματος).

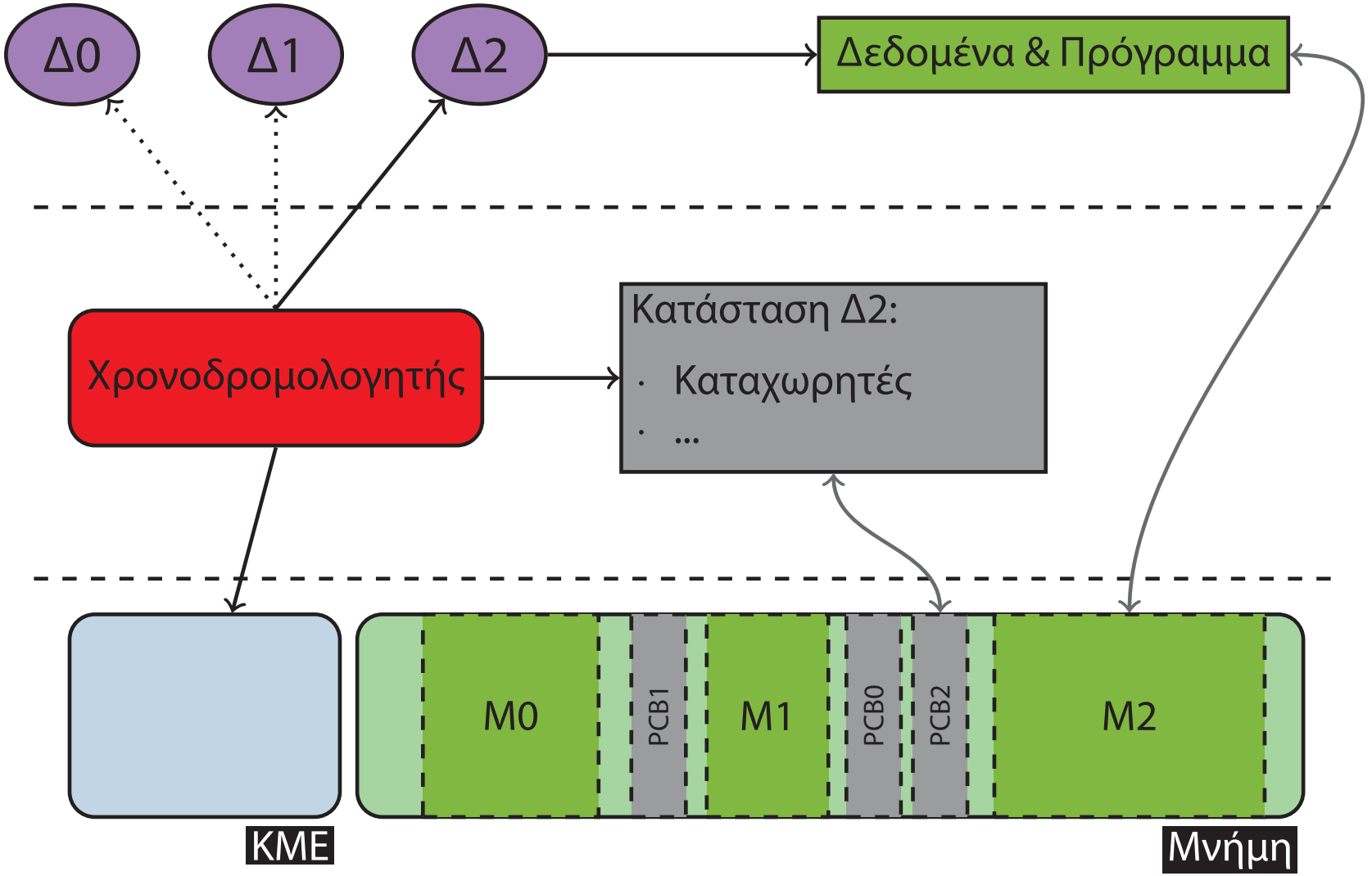
Διεργασίες

Πρόγραμμα που εκτελείται στο Λειτουργικό Σύστημα και βρίσκεται σε συγκεκριμένη κατάσταση.

Διεργασίες (Processes)



Διεργασίες (Processes)



ZeroMyMem()

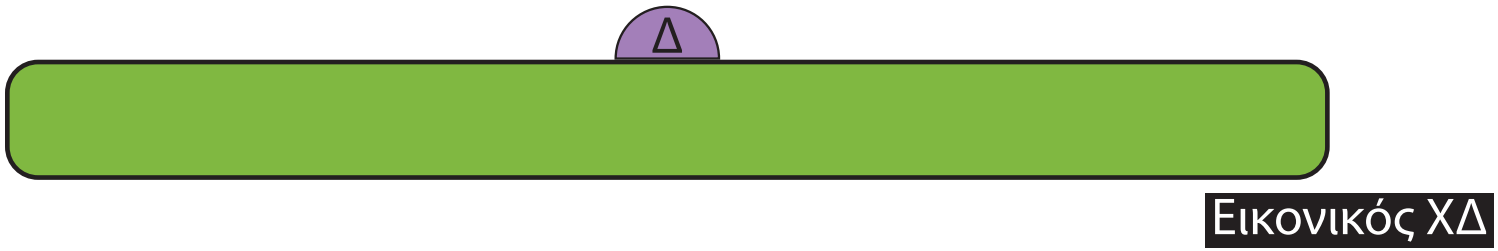
```
/* zero memory [mstart, mend]  */
int ZeroMyMem(char *mstart, char *mend)
{
    unsigned long remaining;
    if (mend <= mstart)
        return -1;
    remaining = mend - mstart;
    do {
        *(mstart + remaining) = 0;
        remaining--;
    } while (remaining >= 0);
    return 0;
}
```

ZeroMyMem()

```
/* zero memory [mstart, mend]  */
int ZeroMyMem(char *mstart, char *mend)
{
    unsigned long remaining;
    if (mend <= mstart)
        return -1;
    remaining = mend - mstart;
    do {
        *(mstart + remaining) = 0;
        remaining--;
    } while (remaining >= 0);
    return 0;
}
```

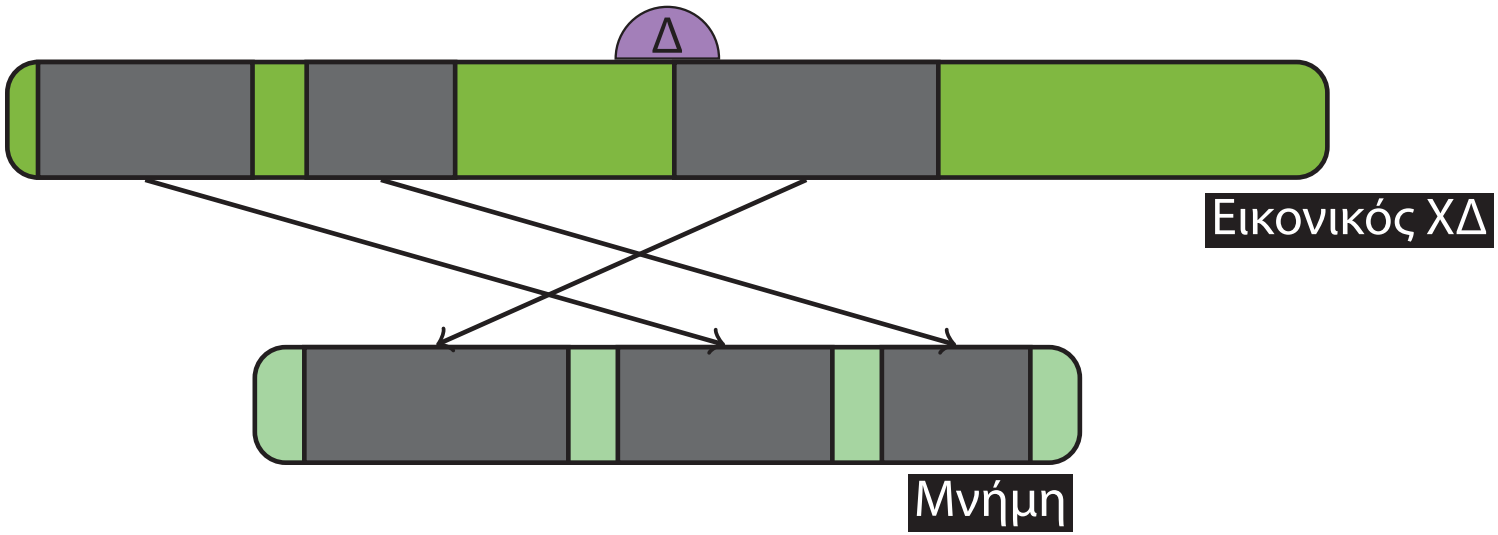
Διαχείριση Μνήμης

- Κάθε διεργασία έχει το δικό της (εικονικό) χώρο διευθύνσεων (ΧΔ)



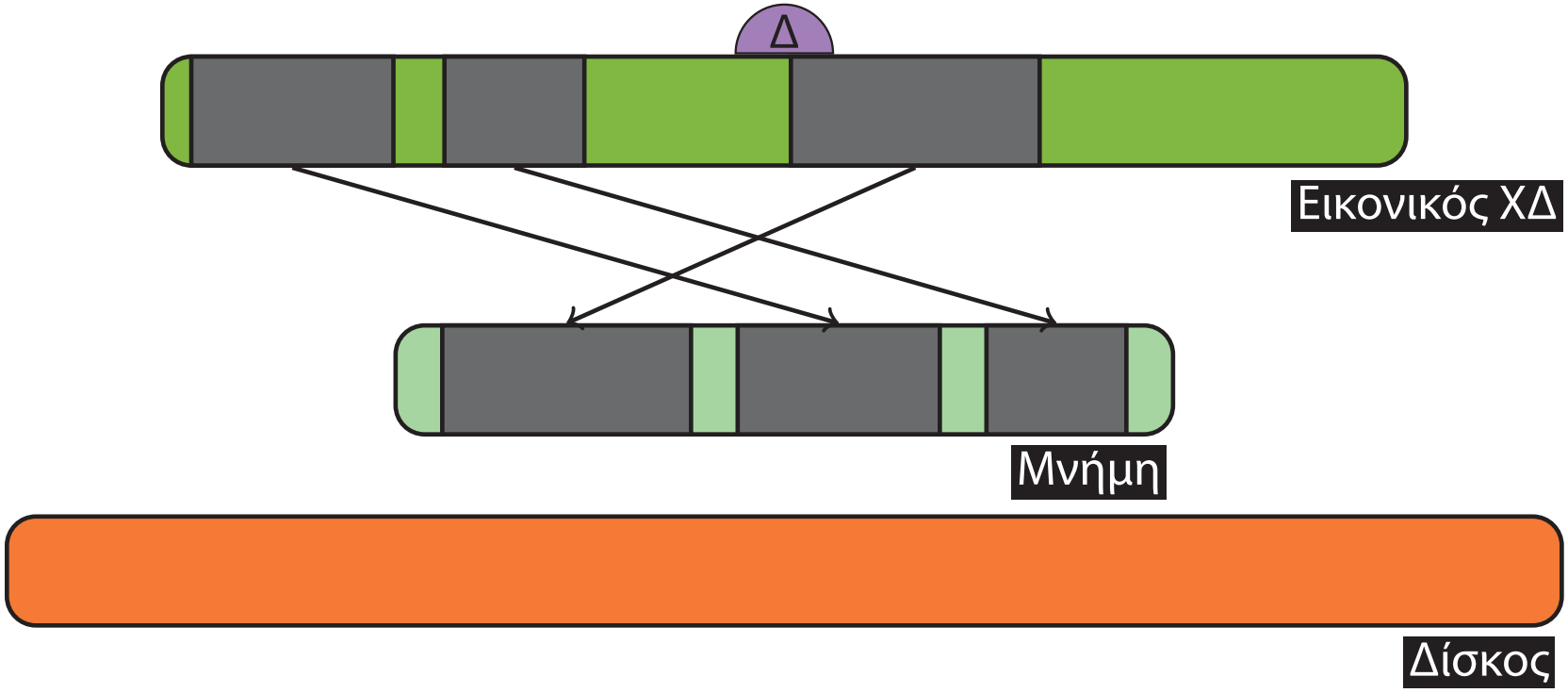
Διαχείριση Μνήμης

- Κάθε διεργασία έχει το δικό της (εικονικό) χώρο διευθύνσεων (ΧΔ)
- Περιοχές του ΧΔ της διεργασίας:
  - Αντιστοιχίζονται σε περιοχές της φυσικής μνήμης



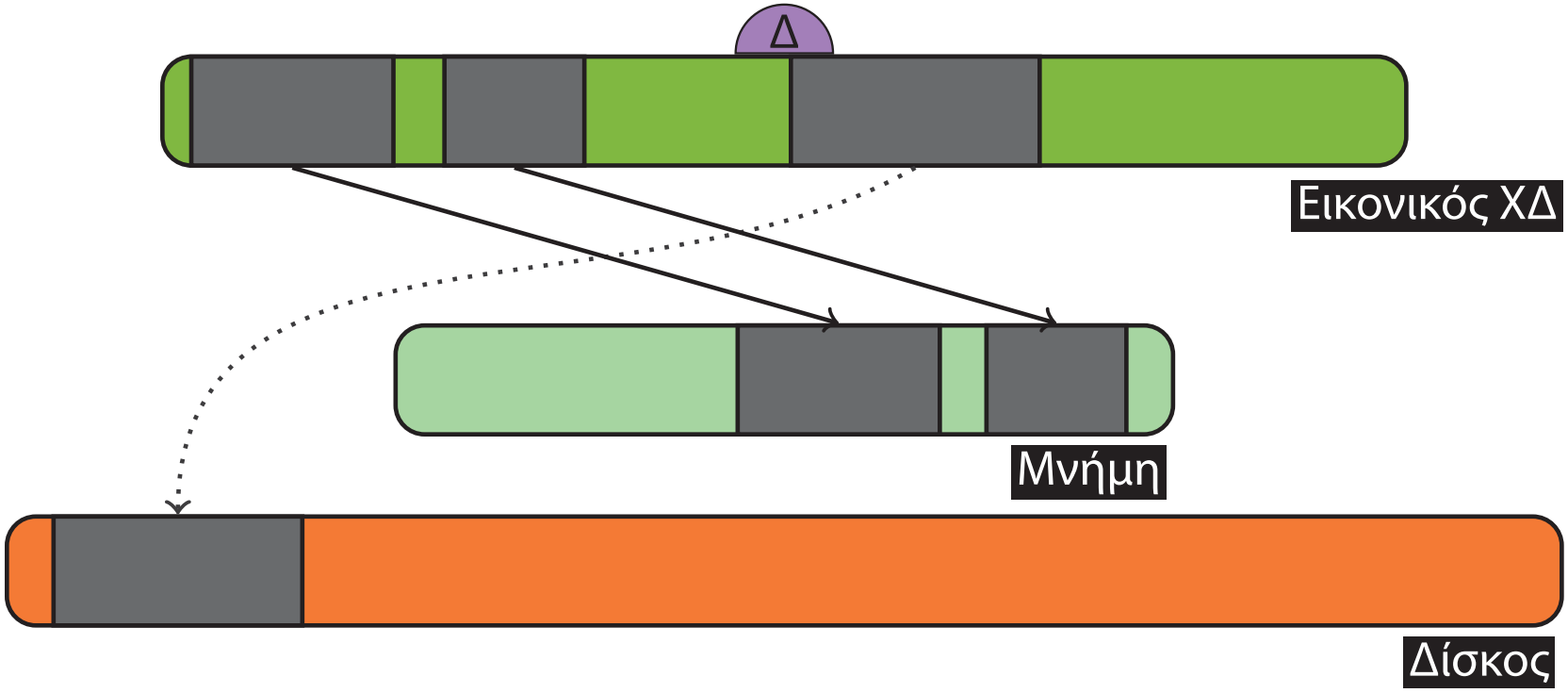
Διαχείριση Μνήμης

- Κάθε διεργασία έχει το δικό της (εικονικό) χώρο διευθύνσεων (ΧΔ)
- Περιοχές του ΧΔ της διεργασίας:
  - Αντιστοιχίζονται σε περιοχές της φυσικής μνήμης
  - Αποθηκεύονται (προσωρινά) σε δευτ. συσκ. αποθήκευσης



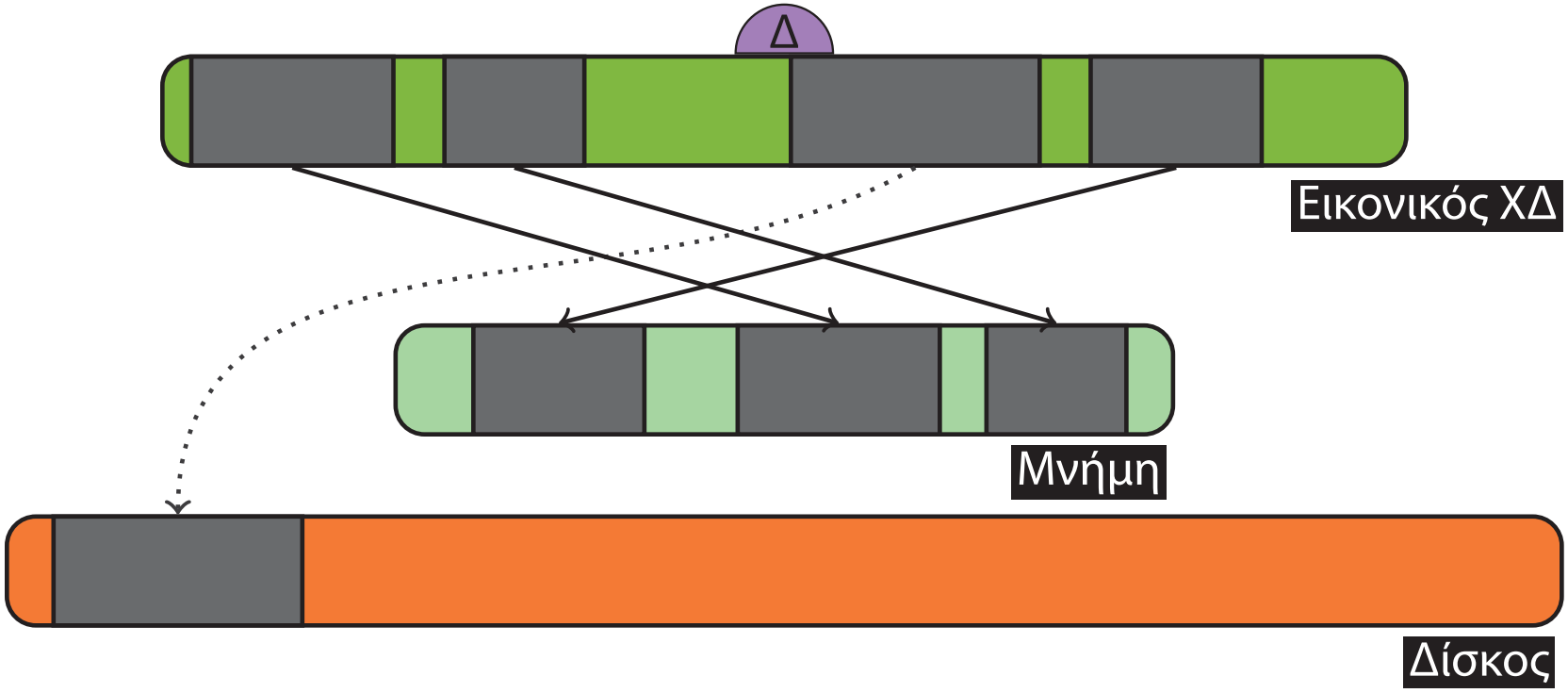
Διαχείριση Μνήμης

- Κάθε διεργασία έχει το δικό της (εικονικό) χώρο διευθύνσεων (ΧΔ)
- Περιοχές του ΧΔ της διεργασίας:
  - Αντιστοιχίζονται σε περιοχές της φυσικής μνήμης
  - Αποθηκεύονται (προσωρινά) σε δευτ. συσκ. αποθήκευσης



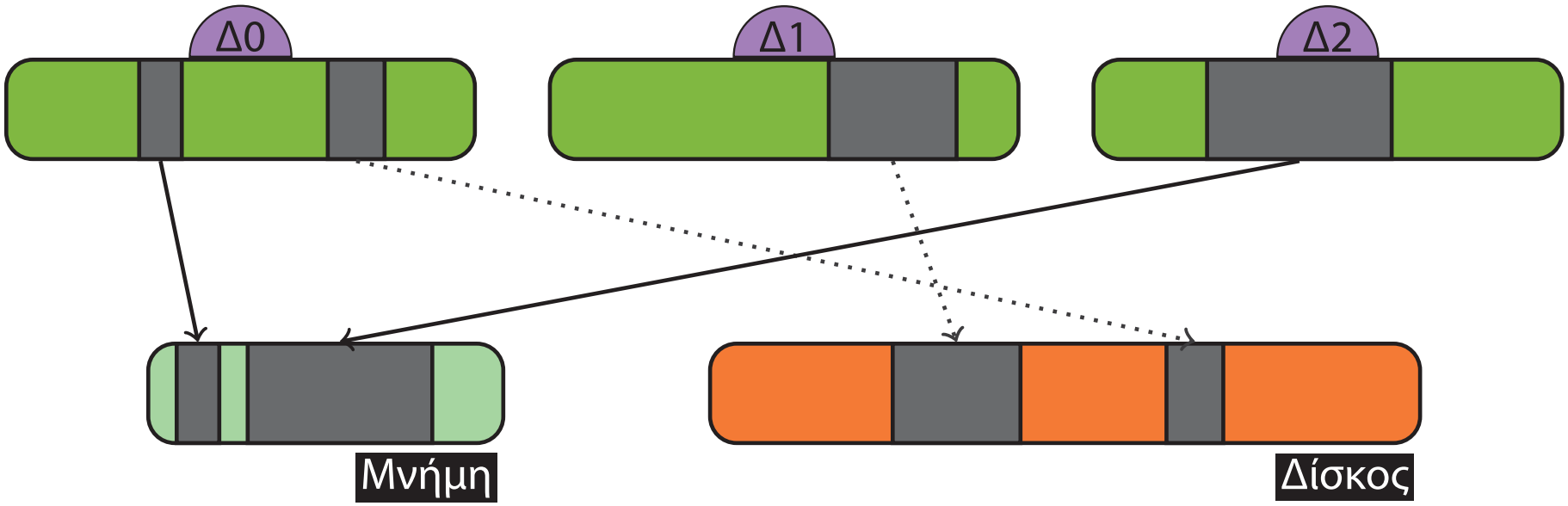
Διαχείριση Μνήμης

- Κάθε διεργασία έχει το δικό της (εικονικό) χώρο διευθύνσεων (ΧΔ)
- Περιοχές του ΧΔ της διεργασίας:
  - Αντιστοιχίζονται σε περιοχές της φυσικής μνήμης
  - Αποθηκεύονται (προσωρινά) σε δευτ. συσκ. αποθήκευσης



Εικονική Μνήμη

- Οι περισσότερες διεργασίες χρησιμοποιούν μόνο μικρό μέρος του Εικονικού ΧΔ τους.
- Όταν δεν επαρκεί η φυσική μνήμη χρησιμοποιείται ο δίσκος ως (προσωρινό) μέσο αποθήκευσης.
- Η χρησιμοποίηση του δίσκου δεν είναι γενικά επιθυμητή, γιατί δημιουργεί προβλήματα επίδοσης



Υλοποίηση Συστήματος Εικονικής Μνήμης

- Υποστήριξη Υλικού για τις αντιστοιχίσεις:
  - `STORE`, `LOAD` εντολές χωρίς την παρέμβαση του ΛΣ.
  - TLB
- Διαχείριση Μνήμης
  - Σελιδοποίηση
  - Αποφυγή Κατακερματισμού
- Πολιτική Swap in/out
  - Βελτιστοποίηση χρήσης μνήμης
  - LRU

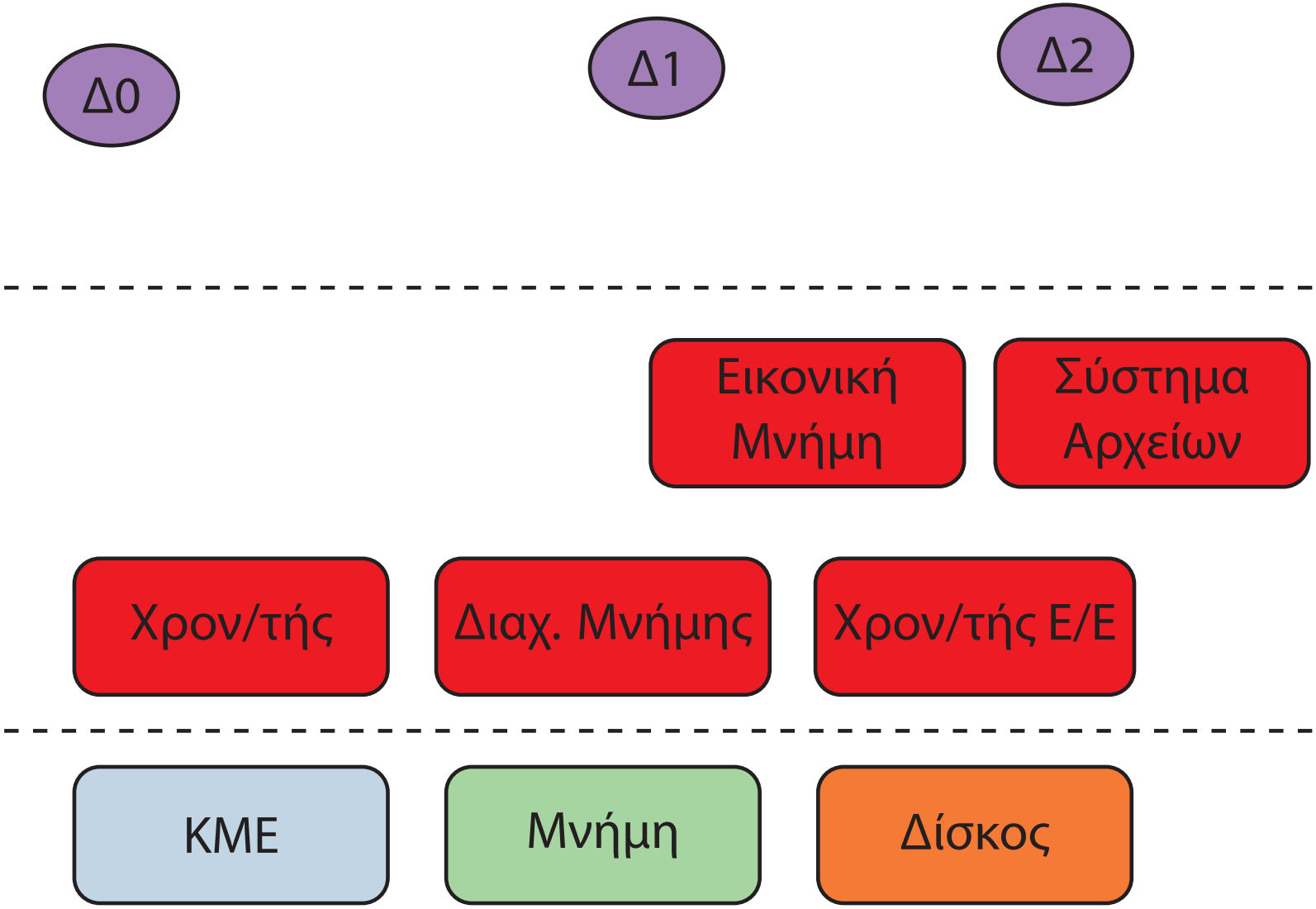
Ανακεφαλαίωση

Το ΛΣ προσφέρει στα προγράμματα μια Εικονική Μηχανή Von Neumann

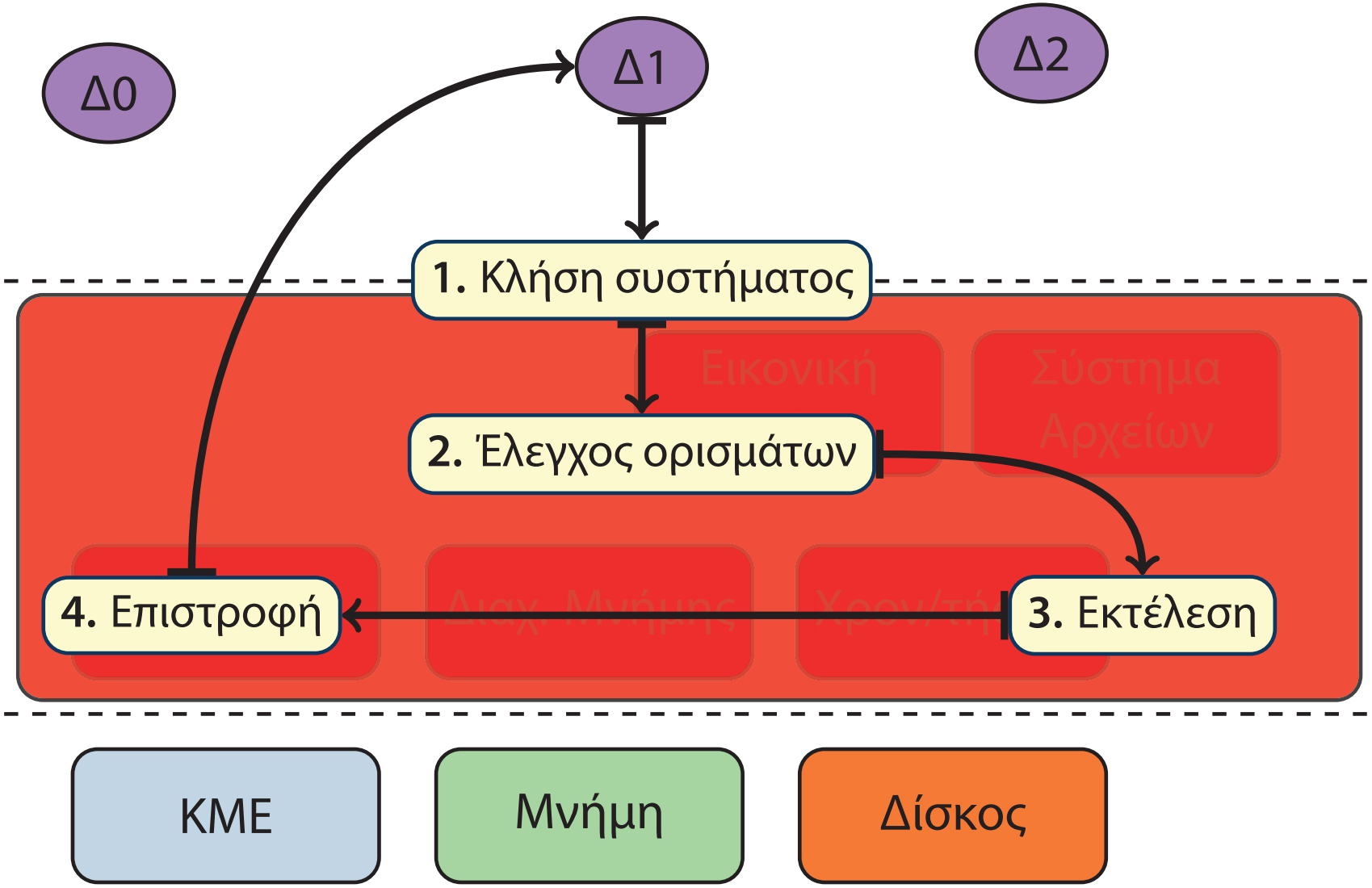
- Διαμοιρασμός χρόνου (Επεξεργαστής)
- Εικονική μνήμη (Μνήμη)

Οι διεργασίες έχουν την (ψευδ-)αίσθηση ότι χρησιμοποιούν αποκλειστικά τον επεξεργαστή και τη μνήμη του συστήματος

Διαχωρισμός Χώρου Χρήστη/Πυρήνα



Διαχωρισμός Χώρου Χρήστη/Πυρήνα

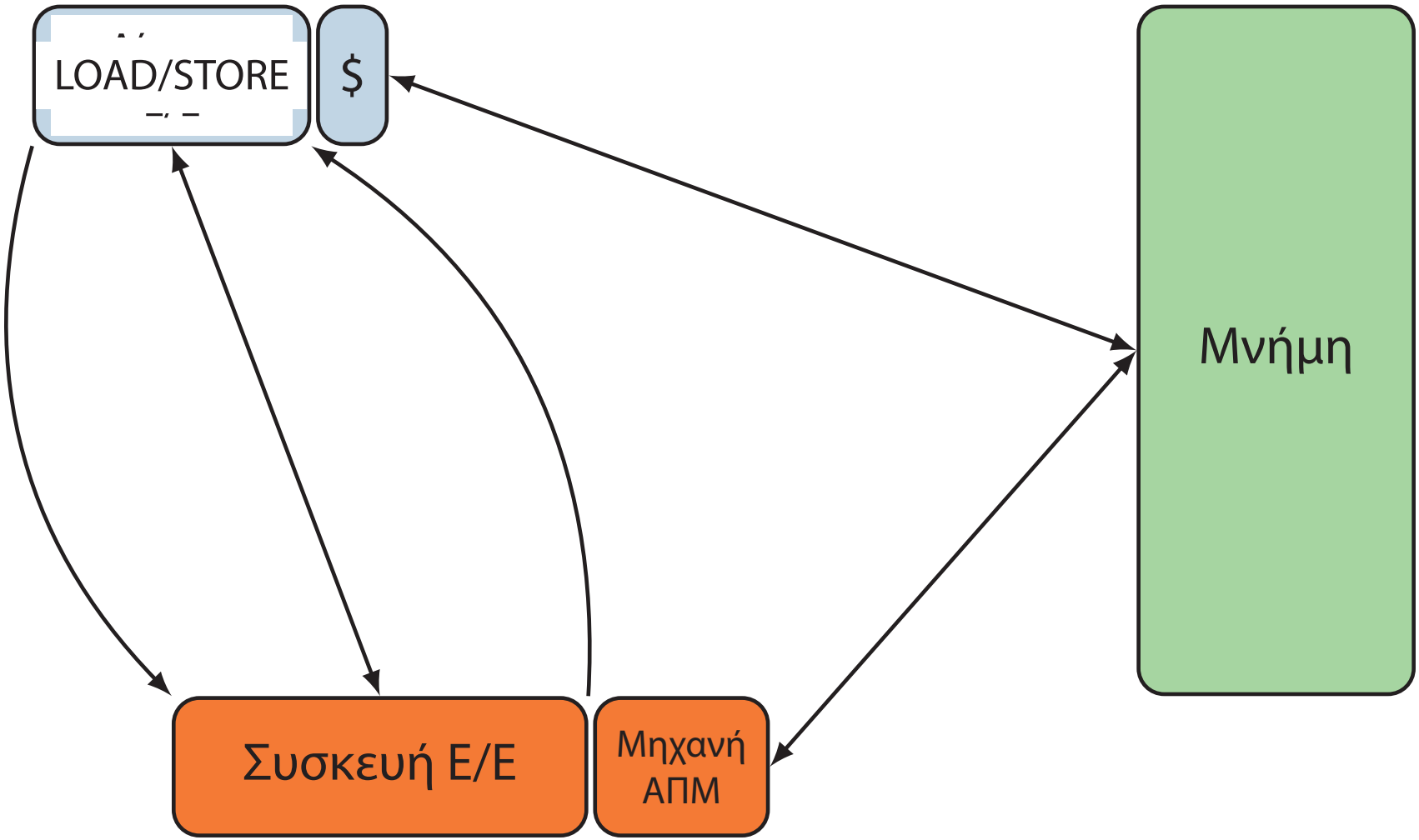


Είσοδος/Έξοδος

- **Όχι** απευθείας πρόσβαση των εφαρμογών στο υλικό
- Κλήσεις Συστήματος (system calls):
  Προγραμματιστική Διεπαφή για τις υπηρεσίες που προσφέρει το ΛΣ στις εφαρμογές
- Υπηρεσίες πάνω από το υλικό
  - Συσκευές Αποθήκευσης (Συστήματα Αρχείων)
  - Προσαρμογείς Δικτύου (Δικτυακά Πρωτόκολλα)
- Πληκτρολόγιο, Οθόνη, Κάμερα, . . .

Ε/Ε Μοντέρνα Υπολογιστικά Συστήματα

(Ασύγχρονο Μοντέλο)



Συσκευές αποθήκευσης

- Μόνιμα δεδομένα (persistent) (σε αντίθεση με μνήμη)
- Σκληροί Δίσκοι
  - Αργή πρόσβαση
  - Χρόνος αναζήτησης (seek time)
  - Solid State Disks (SSDs)
- Γραμμικός χώρος

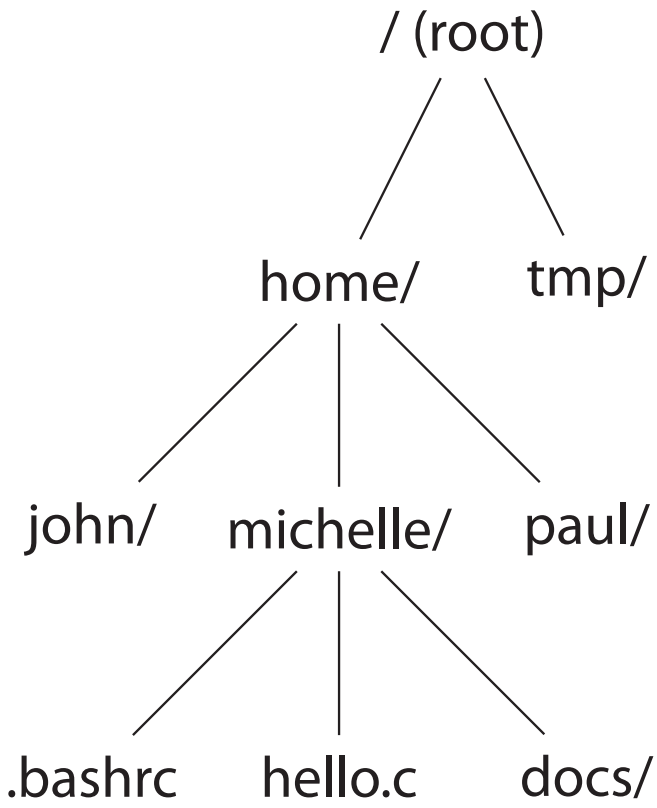
  $\rightarrow$ Όχι ιδανικός για χρήστη/εφαρμογές
  - Συστήματα Αρχείων (Ιεραρχική Δομή)
  - Βάσεις Δεδομένων (SQL)

Συστήματα αρχείων

- Κατάλογοι (Κόμβοι Ιεραρχίας)
- Αρχεία (Δεδομένα)

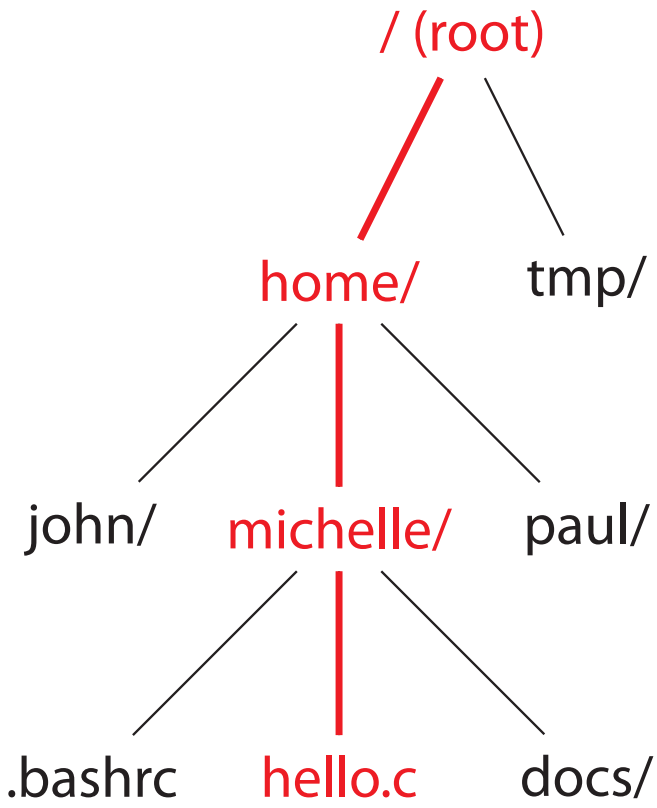
Συστήματα αρχείων

- Κατάλογοι (Κόμβοι Ιεραρχίας)
- Αρχεία (Δεδομένα)



Συστήματα αρχείων

- Κατάλογοι (Κόμβοι Ιεραρχίας)
- Αρχεία (Δεδομένα)



/home/michelle/hello.c

```
#include <stdio.h>

int main(int argc, char **argv)
{
   printf("Hello World!\n");
   return 0;
}
```

Διεπαφή Αρχείων

Ανάγνωση:

- Άνοιγμα (open)
- Ανάγνωση σε μνήμη (read)
- Κλείσιμο (close)

Εγγραφή:

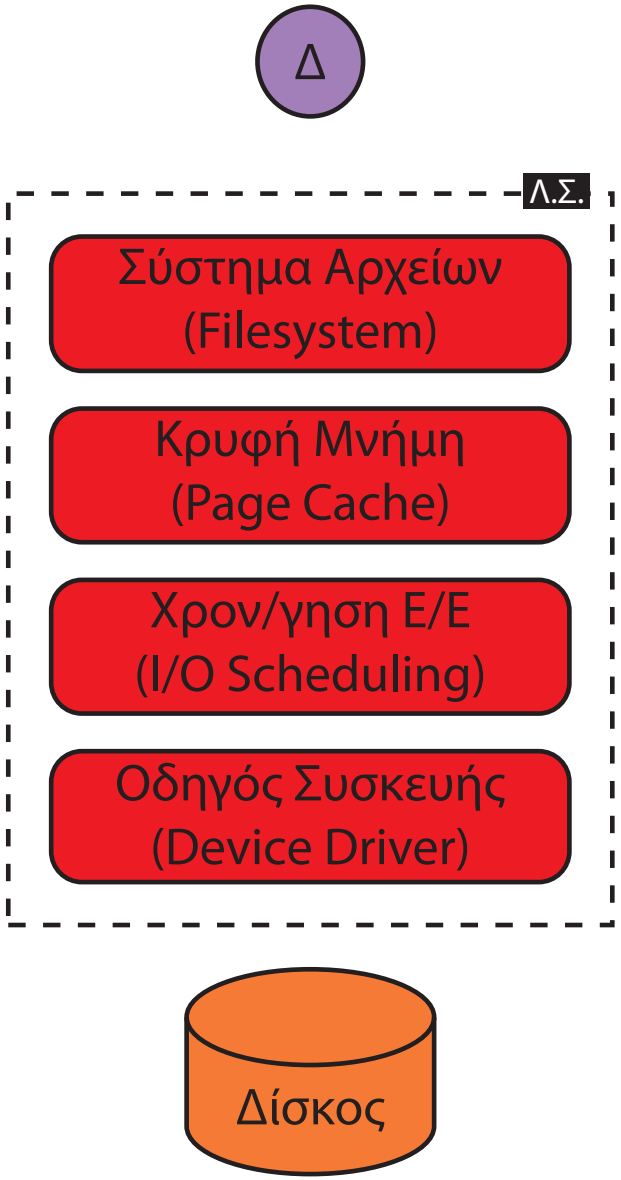
- Άνοιγμα (open)
- Εγγραφή από μνήμη (write)
- Κλείσιμο (close)

Από (εγγραφή) και Προς (ανάγνωση) τη μνήμη

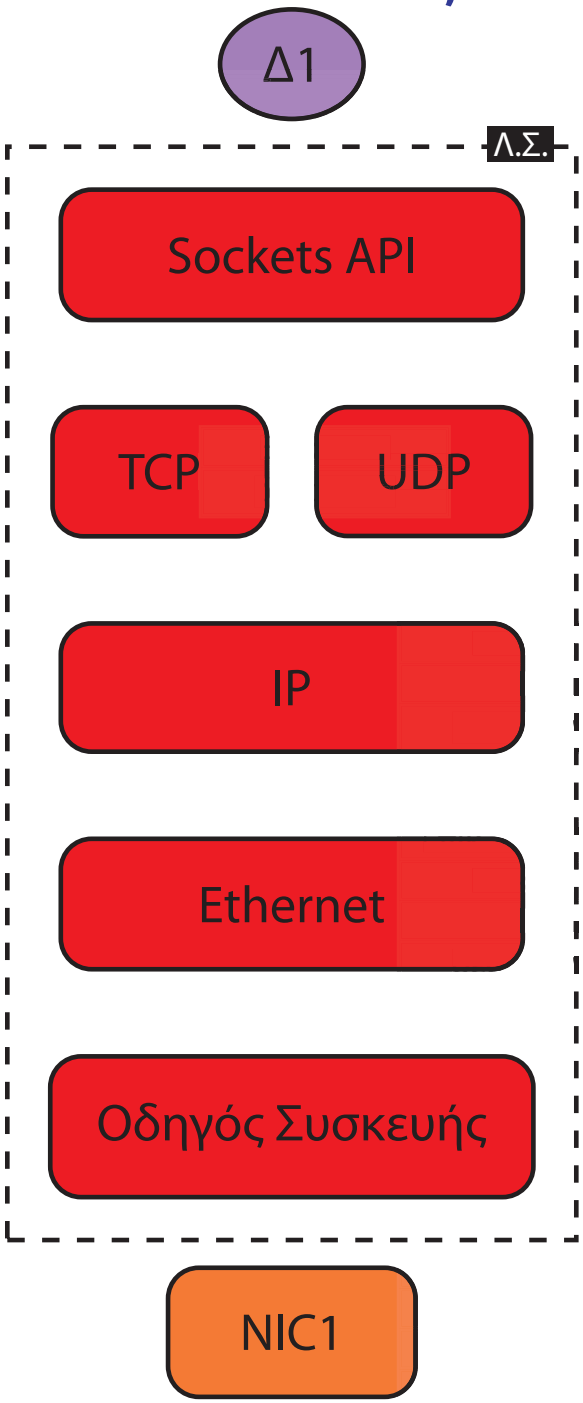
υπάρχουν εξαιρέσεις (πχ `sendfile()`)

ΛΣ και Συσκευές Αποθήκευσης

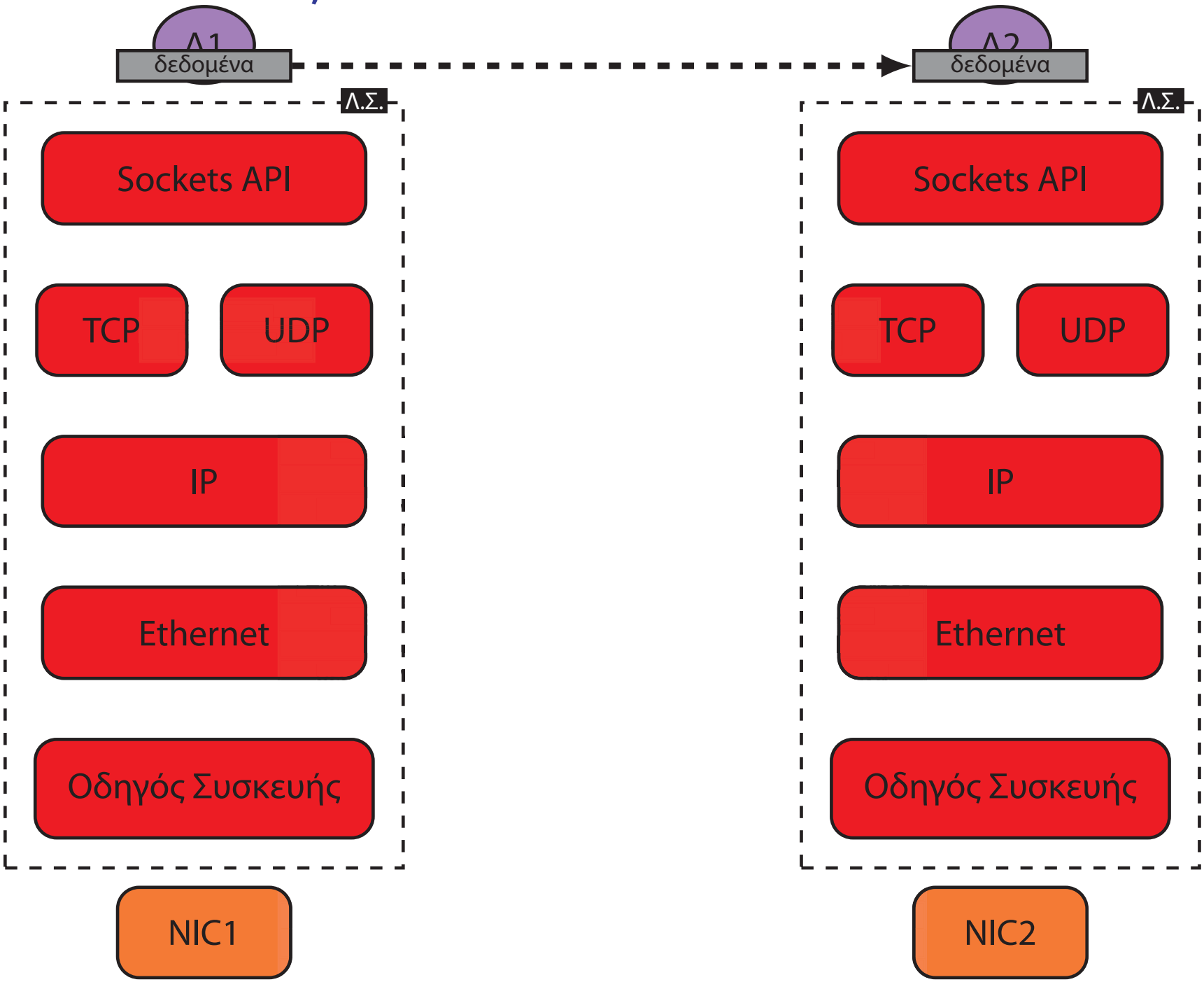
- Συστήματα Αρχείων
  - Ιεραρχική δομή πάνω από γραμμικό χώρο (συσκευή)
- Κρυφή Μνήμη
  - Η πρόσβαση στο δίσκο είναι αργή
  - Περιοχές αρχείων στη μνήμη
- Χρονοδρομολόγηση Ε/Ε
  - Μεγάλος χρόνος αναζήτησης (seek)
  - Βελτιστοποίηση Ε/Ε αιτήσεων



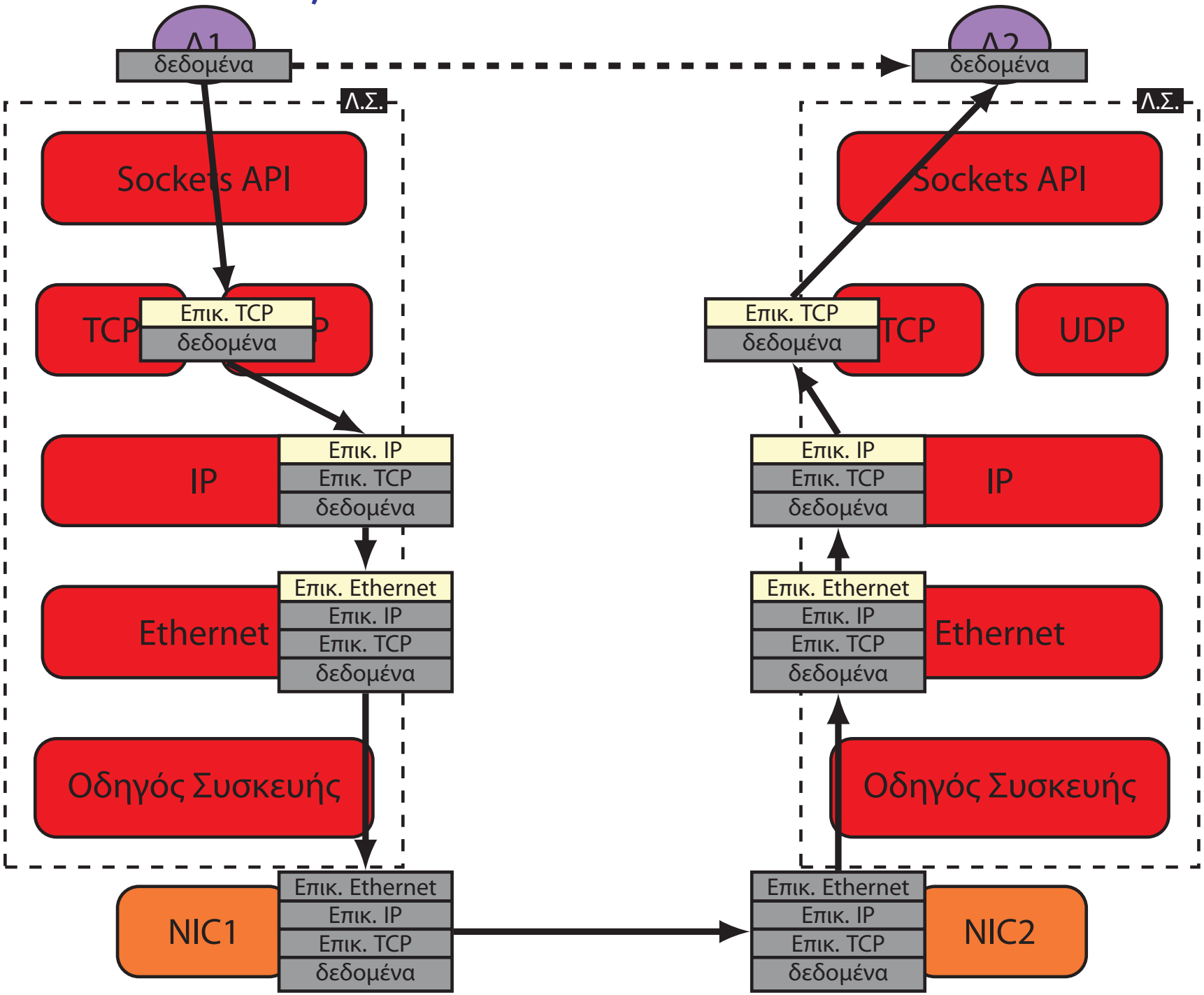
ΛΣ και Συσκευές δικτύου



ΛΣ και Συσκευές δικτύου



ΛΣ και Συσκευές δικτύου



Ανακεφαλαίωση

Διεπαφή πρόσβασης του υλικού για τις εφαρμογές

- Πολλαπλές Διεργασίες (*multitasking*) / Χρήστες (*multiuser*)
  - Προστασία μνήμης / Εικονική μνήμη
  - Διαμοιρασμός επεξεργαστή (time sharing)
  - Περιορισμός εφαρμογών χρήστη (protected mode)
- Συστήματα αρχείων για αποθήκευση δεδομένων
  - Συσκευές Αποθήκευσης (Δίσκοι)
  - Ιεραρχική δομή με καταλόγους και αρχεία
- Επικοινωνία με άλλα υπολογιστικά συστήματα
  - Συσκευές δικτύου (NICs)
  - Στοίβα πρωτοκόλλων επικοινωνίας ({TCP,UDP}/IP)

Εξέλιξη/Διαφοροποίηση

- Τα προηγ. αφορούν κυρίως συστήματα τύπου Desktop/Server
- Τα ΛΣ διαφοροποιούνται ανάλογα με το Υλικό/Ανάγκες
  - Μοναδικός χρήστης / εφαρμογή
    (Πρώτα PCs, Consoles)
  - Batch Systems / Spooling
- Εξαρτώνται από τη χρήση του συστήματος και το υλικό
  - Ενσωματωμένα συστήματα
  - iPhone: ένα πρόγραμμα τη φορά, ένας χρήστης

Εικονικές Μηχανές

Virtual Machines

Υλοποίηση σε λογισμικό ενός υπολογιστικού συστήματος, που εκτελεί προγράμματα σαν να είναι φυσική μηχανή

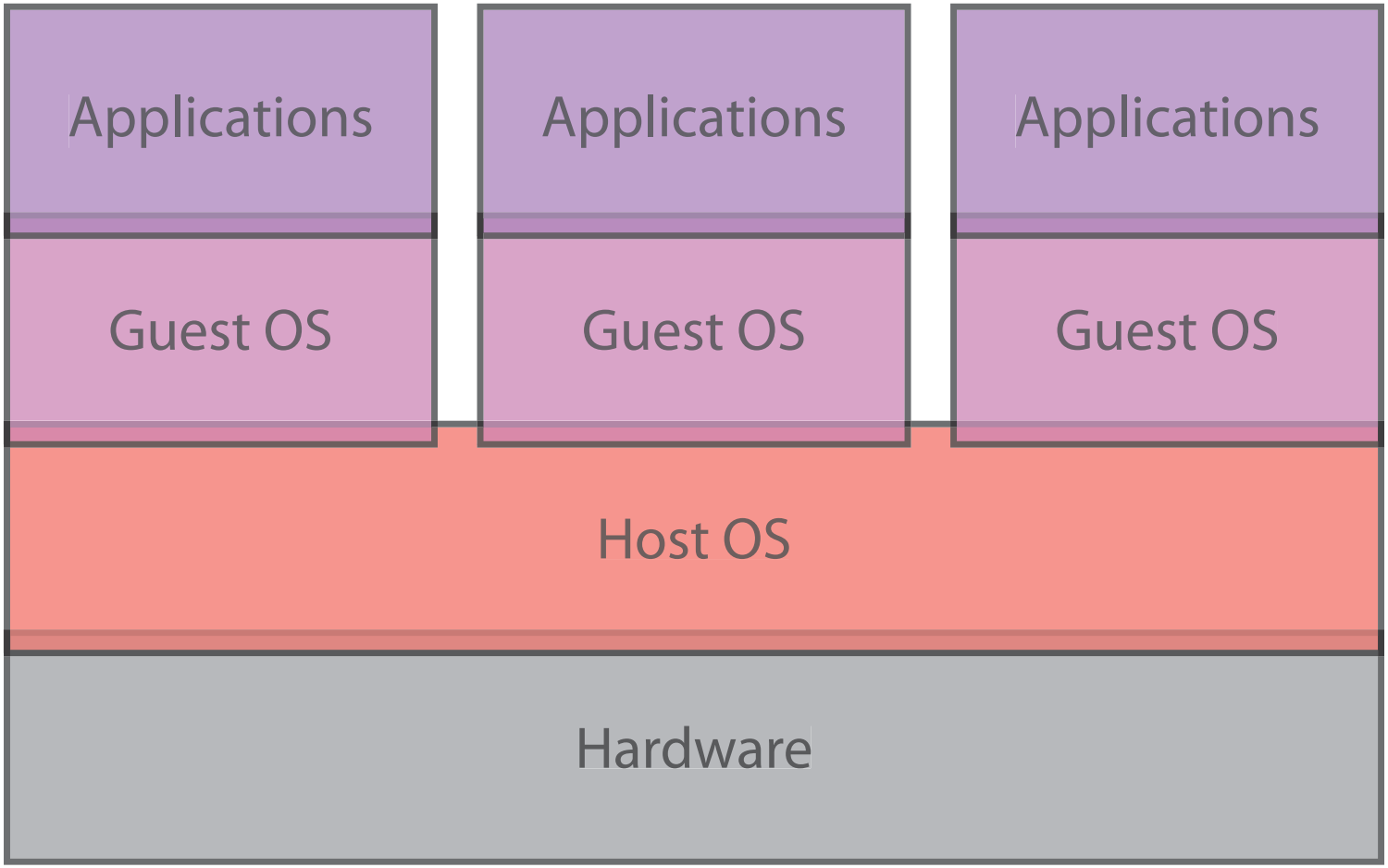
- Εικονικές Μηχανές Συστημάτων (Xen, Virtual Box, VMware)
- Εικονικές Μηχανές Διεργασιών (Java VM, .NET CLR)

Εικονικές Μηχανές Συστημάτων

- Το ΛΣ που τρέχει στο πραγματικό υλικό και παρέχει την λειτουργικότητα των Εικονικών Μηχανών ονομάζεται *host OS*
- Το ΛΣ που εκτελείται μέσα στις Εικονικές Μηχανές ονομάζεται *guest OS*

Οι πόροι του συστήματος μοιράζονται στις Εικονικές Μηχανές (συνήθως με στατικό τρόπο), από το host OS. Οι πόροι κάθε Εικονικής Μηχανής είναι υπό τη διαχείριση του guest OS.

Παράδειγμα Εικονικής Μηχανής Συστήματος



Βιβλιογραφία (1/3)

- Operating System Concepts,
  Abraham Silberschatz, Peter Baer Galvin, Greg Gagne,
  Wiley 8th edition, 2008.
- Operating Systems: Internals and Design Principles
  William Stallings,
  Prentice Hall, 6th edition, 2008.
- Modern Operating Systems,
  Andrew S Tannenbaum, Prentice Hall,
  3rd edition, 2007.
- Operating Systems Design and Implementation,
  A. S. Tannenbaum, A. S. Woodhull,
  Prentice Hall, 3rd edition, 2007.
- The Design of the Unix Operating System,
  M. Bach,
  Prentice Hall, 1986.

Βιβλιογραφία (2/3)

- The Design and Implementation of the 4.4 BSD Operating System, Marshall Kirk McKusick, Keith Bostic, Michael J. Karels, John S. Quarterman,
  Addison-Wesley Professional, 2nd edition,1996.
- The Design and Implementation of the FreeBSD Operating System, Marshall Kirk McKusick, George V. Neville-Neil, Addison-Wesley Professional,
  1st edition, 2004.
- The Magic Garden Explained: The Internals of Unix System V Release 4: An Open Systems Design,
  Berny Goodheart, James Cox,
  Prentice Hall, 1995.
- UNIX Internals: The New Frontiers,
  Uresh Vahalia,
  Prentice Hall, 1995.

Βιβλιογραφία (3/3)

- Understanding the Linux Kernel,
  Daniel Bovet, Marco Cesati,
  O' Reilly Media, 3rd edition, 2005.
- Linux Device Drivers,
  Jonathan Corbet, Alessandro Rubin, Greg Kroah-Hartman,
  O'Reilly Media, 3rd Edition, 2005.
- Understanding Linux Network Internals
  Christian Benvenuti, Marco Cesati,
  O'Reilly Media, 2005.