

Εισαγωγή στα Λειτουργικά Συστήματα

Λειτουργικά Συστήματα

6ο εξάμηνο ΣΗΜΜΥ

ακ. έτος 2022-2023

<https://helios.ntua.gr/user/index.php?id=1021>



Διδάσκοντες: Νεκτάριος Κοζύρης nkoziris@cslab.ece.ntua.gr
Γιώργος Γκούμας goumas@cslab.ece.ntua.gr

Εργαστήριο Υπολογιστικών Συστημάτων
ΕΜΠ

Μάρτιος 2023

Τμήματα

- ▶ Το μάθημα διεξάγεται σε δύο Τμήματα:
 - ▶ Τμήμα 1 (Ν. Κοζύρης - Γκούμας)
 - ▶ Τμήμα 2 (Π. Τσανάκας)
- ▶ Μπορείτε να επιλέξετε ελεύθερα σε ποιο τμήμα θα παρακολουθήσετε
- ▶ Τα δύο τμήματα έχουν ξεχωριστά:
 - ▶ Εργαστηριακό μέρος
 - ▶ Τελική εξέταση

Μαθήματα

- ▶ Δευτέρα 15:15 - 17:00 (Αμφ. 1)
- ▶ Σε έκτακτες περιπτώσεις αναπληρώσεις 17:15-18:00
 - ▶ Θεωρία Μαθήματος
 - ▶ Παρουσιάσεις εργαστηριακών ασκήσεων
- ▶ Τρίτη: Εργαστήριο (δια ζώσης)
 - ▶ Συζητήσεις επί των εργαστηριακών ασκήσεων
 - ▶ Υλοποίηση ασκήσεων
 - ▶ Επίλυση αποριών
 - ▶ Επίδειξη ασκήσεων
 - ▶ Εξοικείωση με τα εργαλεία χρήστη του Linux (hands-on)

Μάθημα

- ▶ Βαθμολογία:
 - ▶ 5 Βαθμοί η τελική εξέταση
 - ▶ 5 Βαθμοί το εργαστήριο
 - ▶ Απαιτείται προβιβάσιμος βαθμός στο γραπτό
 - ▶ **ΠΡΟΣΟΧΗ!:** Σε περιπτώσεις μεγάλων αποκλίσεων ανάμεσα στο βαθμό εργαστηρίου και γραπτού, δίνεται μεγαλύτερη βαρύτητα στο γραπτό.
- ▶ Τελική Εξέταση:
 - ▶ Κλειστά βιβλία και σημειώσεις
 - ▶ Ένα Α4 φύλλο (σκονάκι)
- ▶ Απαιτείται:
 - ▶ Παρουσία και συμμετοχή στο εργαστήριο
 - ▶ Παρακολούθηση και ενασχόληση με το μάθημα (όχι 1 εβδομάδα πριν την εξεταστική)

Εργαστήριο

- ▶ Ομάδες των δύο ατόμων
- ▶ Εργαστηριακές Ασκήσεις: Προγραμματισμός σε περιβάλλον Linux
- ▶ Εργαστηριακές βάρδιες κάθε Τρίτη:
 - ▶ 10:45-12:30
 - ▶ 12:45-14:30
- ▶ Εξέταση ασκήσεων: ασκήσεις που δεν εξετάζονται εμπρόθεσμα θα έχουν βαθμολογική επιβάρυνση (10% του βαθμού για κάθε εβδομάδα καθυστέρησης).

Εγγραφές για το μάθημα

1. Εγγραφή στο τμήμα Ν. Κοζύρη / Γ. Γκούμα (εφόσον το επιλέξετε): Κάνετε enrol στην ιστοσελίδα του μαθήματος (Τμήμα 1):
<https://helios.ntua.gr/user/index.php?id=1021>
2. Εγγραφή στο εργαστήριο (οδηγίες θα δοθούν στην ιστοσελίδα του μαθήματος)
3. Εγγραφή στη mailing list του μαθήματος (οδηγίες στην ιστοσελίδα του μαθήματος)

Επικοινωνία

- ▶ Ιστοσελίδα μαθήματος:
<https://helios.ntua.gr/user/index.php?id=1021>
(τρέχουσα)
www.cslab.ece.ntua.gr/courses/os (προηγ. ακ. έτη)
- ▶ Mailing List:
lists.cslab.ece.ntua.gr/mailman/listinfo/os
Απαραίτητη προϋπόθεση η εγγραφή σας στη λίστα! Η διαδικασία εγγραφής περιλαμβάνει αυτόματο mail επιβεβαίωσης που στέλνεται από το σύστημα. Σε περίπτωση που δεν το λάβετε ελέγξτε και το φάκελο Junk του email client σας.
- ▶ Mail διδασκόντων:
{goumas,nkoziris}@cslab.ece.ntua.gr
- ▶ Mail βοηθών εργαστηρίου:
oslab-assist@cslab.ece.ntua.gr

Καλές πρακτικές χρήσης της λίστας του μαθήματος

- ▶ Η λίστα ενδείκνυται για επίλυση αποριών. Επικοινωνούμε απευθείας με τους διδάσκοντες για διαδικαστικά θέματα του μαθήματος και με τους υπεύθυνους του εργαστηρίου για διαδικαστικά θέματα εργαστηρίου (π.χ. απουσία, μη τήρηση προθεσμίας, κλπ).
- ▶ Κάνουμε μια μικρή αναζήτηση στο διαδίκτυο για να επιβεβαιώσουμε ότι η απορία μας δεν έχει απλή απάντηση. Προαιρετικά μπορούμε να αναζητήσουμε απάντηση και στο αρχείο της λίστας <http://lists.cslab.ece.ntua.gr/mailman/private/os/>.
- ▶ Δεν απαντάμε σε απορίες συμφοιτητών μας και δεν ανοίγουμε θέματα συζήτησης μεταξύ των φοιτητών.
- ▶ Καταβάλουμε προσπάθεια οι απορίες μας να είναι συνοπτικά διατυπωμένες και ο λόγος μας καλά δομημένος.
- ▶ Δεν στέλνουμε κώδικα από εργασίες σε εξέλιξη.
- ▶ **Έχοντας λάβει υπόψη τα παραπάνω, δε διστάζουμε να στείλουμε mail στη λίστα!**

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.

Περιεχόμενα μαθήματος

- ▶ Εισαγωγή
- ▶ Διεργασίες & Νήματα
- ▶ Χρονοδρομολόγηση
- ▶ Συγχρονισμός
- ▶ Διαχείριση μνήμης και σύστημα εικονικής μνήμης
- ▶ Συστήματα αρχείων
- ▶ Χρονοδρομολόγηση E/E
- ▶ Εικονικές Μηχανές

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

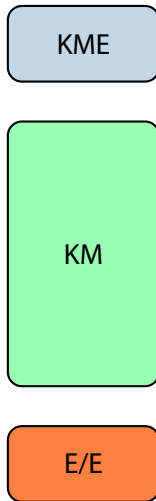
- ▶ Σχέση Εξάρτησης μεταξύ ΛΣ και αρχιτεκτονικής
- ▶ Πολλές λειτουργίες των ΛΣ χρειάζονται υποστήριξη από το Υλικό
- ▶ Νέες δυνατότητες του Υλικού χρειάζονται υποστήριξη από το ΛΣ, ώστε να είναι διαθέσιμες στις εφαρμογές

Τί γνωρίζουμε / Τί θα μάθουμε

- ▶ Οργάνωση και σχεδίαση υπολογιστών
 - ▶ Instruction Set Architecture
 - ▶ Προγραμματισμό σε Assembly
 - ▶ Μικροαρχιτεκτονική
- ▶ Εκτέλεση ενός προγράμματος στον επεξεργαστή
 - ▶ Κυρίως χρήση CPU μνήμης
 - ▶ Βασικές λειτουργίες E/E
- ▶ Πώς μπορούν να εκτελεστούν δύο προγράμματα ταυτόχρονα;
- ▶ Πώς μπορούν να χρησιμοποιούν δύο χρήστες το ίδιο υπολογιστικό σύστημα;
- ▶ Πώς θα διαμοιραστούν πολλαπλά προγράμματα και χρήστες τους πεπερασμένους πόρους του υπολογιστικού συστήματος;
- ▶ Πώς θα απομονώσω μεταξύ τους τα προγράμματα που εκτελούνται;
- ▶ Πώς θα γίνεται η πρόσβαση στις περιφερειακές συσκευές;
- ▶ Πώς διαβάζουν τα προγράμματα αρχεία από το σκληρό δίσκο;

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

KME

%a : undef
%pc: START

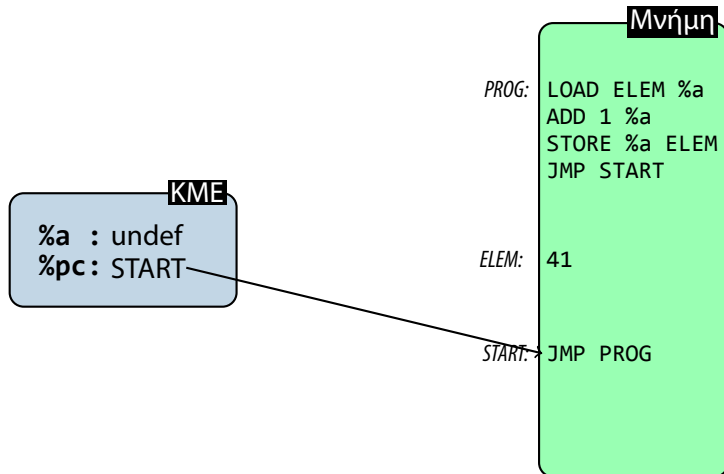
Μνήμη

PROG: LOAD ELEM %a
ADD 1 %a
STORE %a ELEM
JMP START

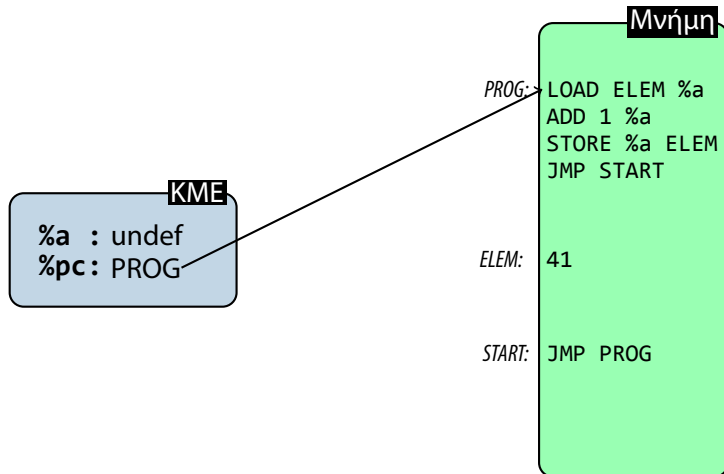
ELEM: 41

START: JMP PROG

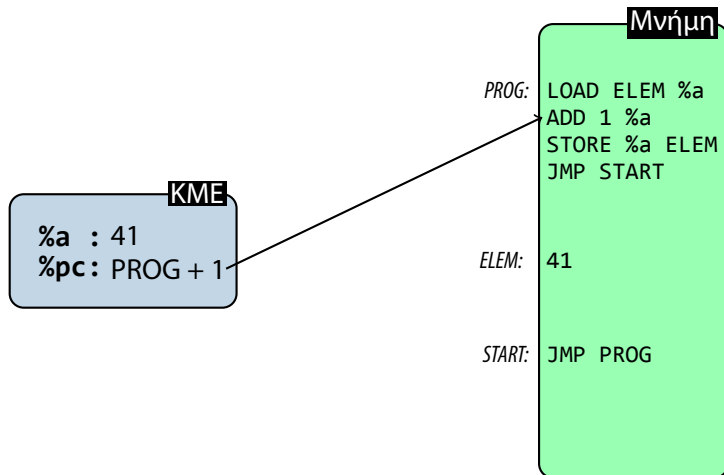
Μοντέλο Εκτέλεσης Προγραμμάτων



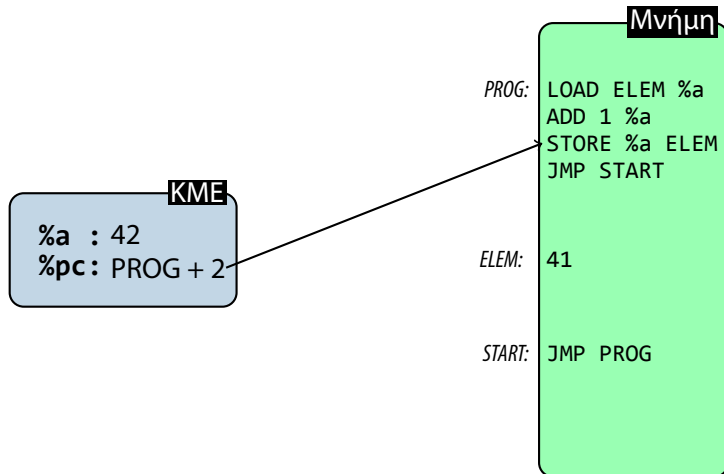
Μοντέλο Εκτέλεσης Προγραμμάτων



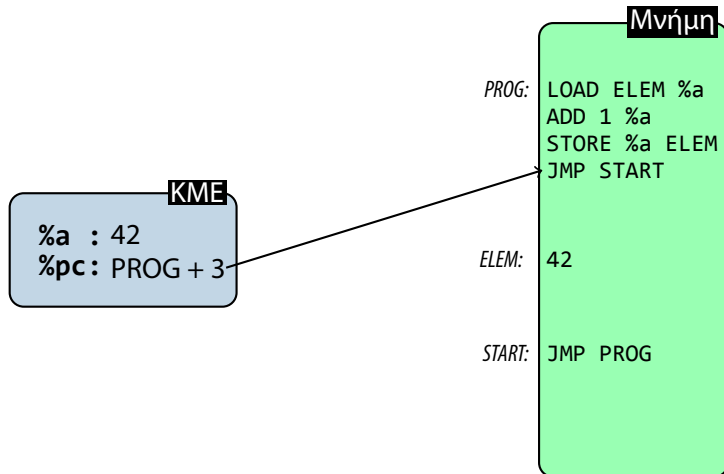
Μοντέλο Εκτέλεσης Προγραμμάτων



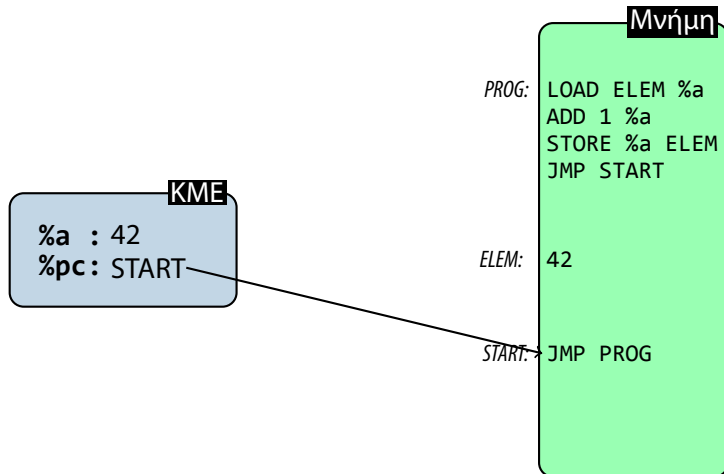
Μοντέλο Εκτέλεσης Προγραμμάτων



Μοντέλο Εκτέλεσης Προγραμμάτων



Μοντέλο Εκτέλεσης Προγραμμάτων



Προβλήματα (προηγ.) Μοντέλου Εκτέλεσης

?

Προβλήματα (προηγ.) Μοντέλου Εκτέλεσης

Εκτέλεση πολλαπλών προγραμμάτων

Προβλήματα (προηγ.) Μοντέλου Εκτέλεσης

Εκτέλεση πολλαπλών προγραμμάτων

Διαμοιρασμός χρόνου

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

Προβλήματα (προηγ.) Μοντέλου Εκτέλεσης

Εκτέλεση πολλαπλών προγραμμάτων

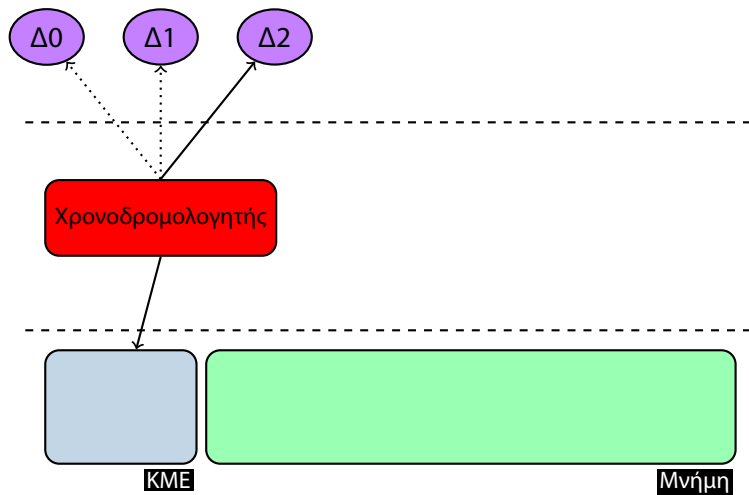
Διαμοιρασμός χρόνου

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

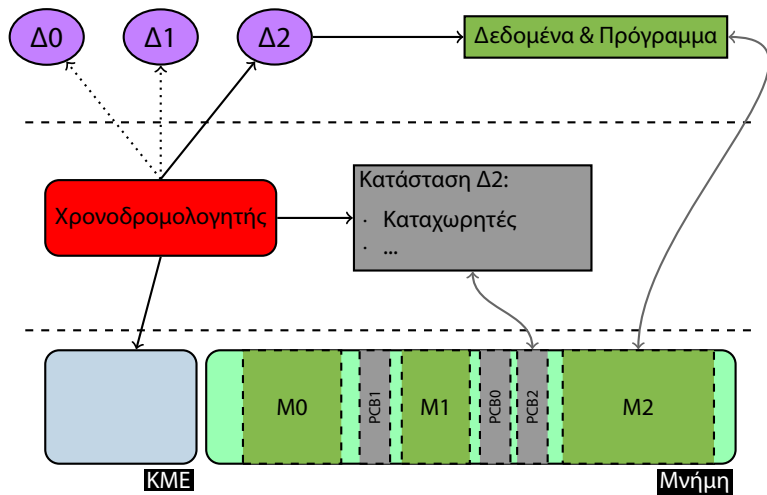
Διεργασίες

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()

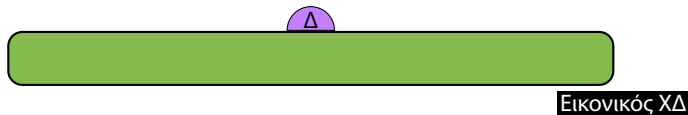
```
/* zero memory [mstart, mend] */
int ZeroMyMem(char *mstart, char *mend)
{
    unsigned long remaining;
    if (mend <= mstart)
        return -1;
    remaining = mend - mstart;
    do {
        *(mstart + remaining) = 0;
        remaining--;
    } while (remaining >= 0);
    return 0;
}
```

Συγχρονισμός

- ▶ Πολλές διεργασίες μπορεί να έχουν πρόσβαση σε κοινούς πόρους (π.χ. κοινές μεταβλητές στη μνήμη)
- ▶ Αυτό μπορεί να δημιουργήσει πρόβλημα (π.χ. counter++)
- ▶ Ποιο είναι το πρόβλημα του συγχρονισμού;
- ▶ Πώς μπορεί να συνεισφέρει το ΛΣ (μαζί με άλλα επίπεδα του υπολογιστικού συστήματος) στην επίλυσή του;

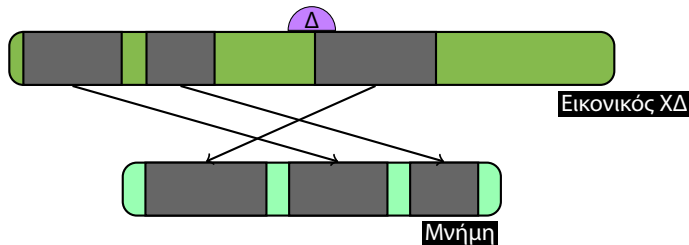
Διαχείριση Μνήμης

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



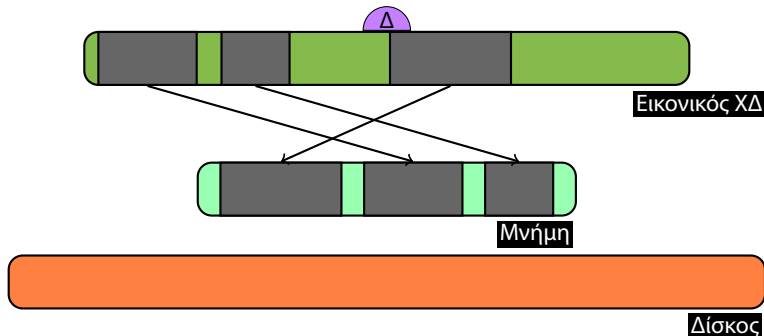
Διαχείριση Μνήμης

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



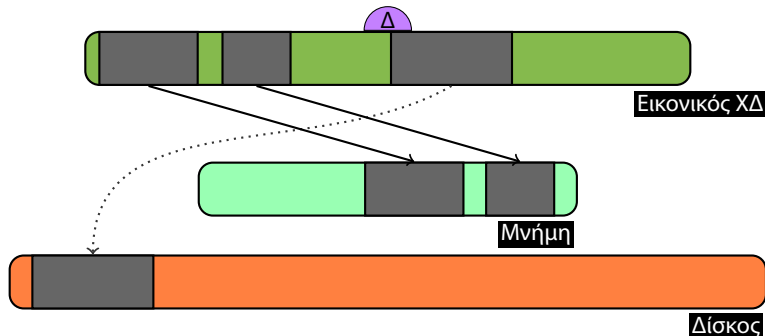
Διαχείριση Μνήμης

- 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



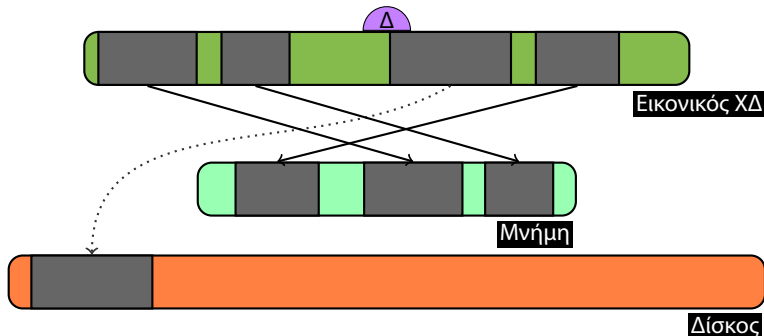
Διαχείριση Μνήμης

- 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



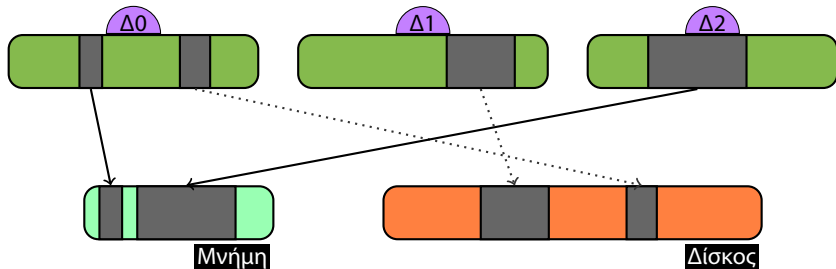
Διαχείριση Μνήμης

- 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

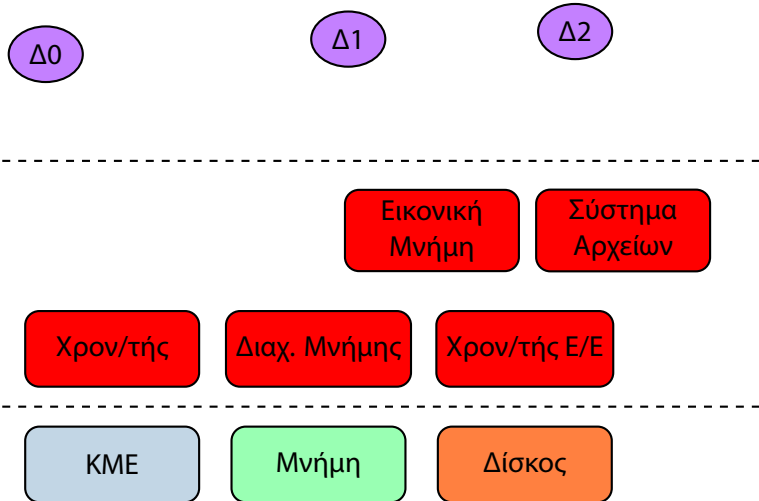
Ανακεφαλαίωση

Το ΛΣ προσφέρει στα προγράμματα μια Εικονική Μηχανή Von Neumann

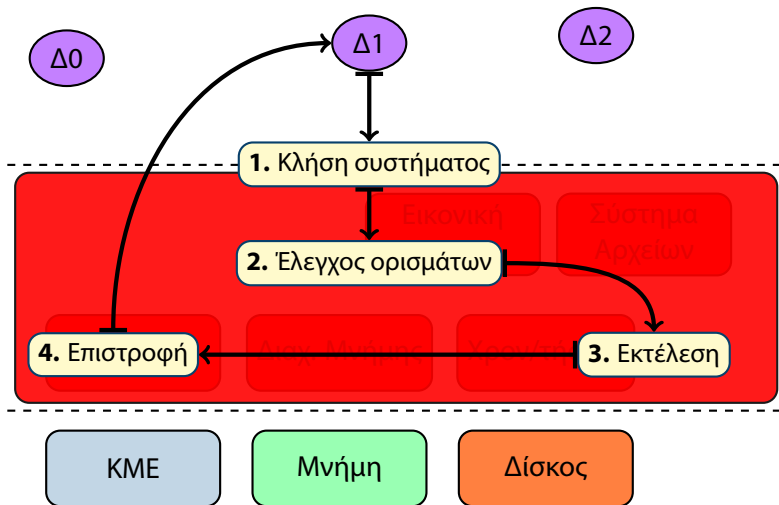
- ▶ Διαμοιρασμός χρόνου (Επεξεργαστής)
- ▶ Εικονική μνήμη (Μνήμη)

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

Διαχωρισμός Χώρου Χρήστη/Πυρήνα



Διαχωρισμός Χώρου Χρήστη/Πυρήνα



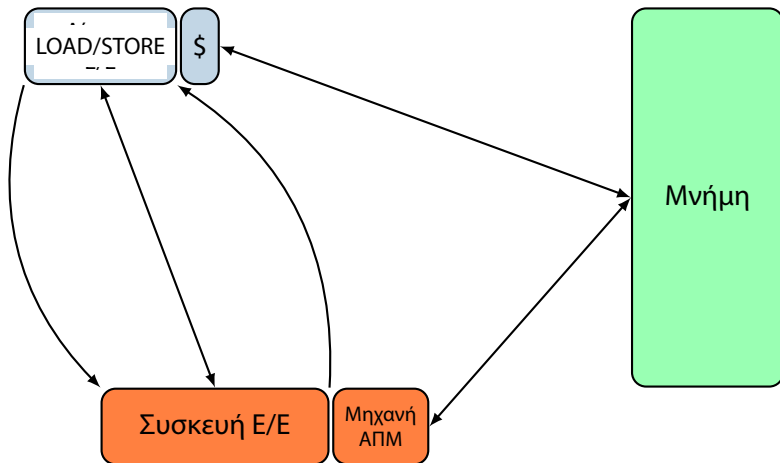
Είσοδος/Έξοδος

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

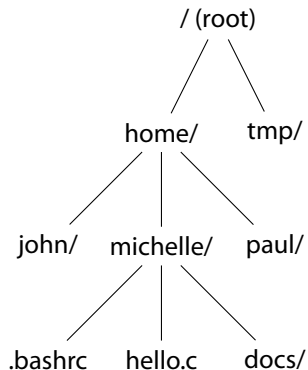
- Μόνιμα δεδομένα (persistent) (σε αντίθεση με μνήμη)
- Σκληροί Δίσκοι
 - Αργή πρόσβαση
 - Χρόνος αναζήτησης (seek time)
 - Solid State Disks (SSDs)
- Γραμμικός χώρος
 - Όχι ιδανικός για χρήστη/εφαρμογές
 - Συστήματα Αρχείων (Ιεραρχική Δομή)
 - Βάσεις Δεδομένων (SQL)

Συστήματα αρχείων

- Κατάλογοι (Κόμβοι Ιεραρχίας)
- Αρχεία (Δεδομένα)

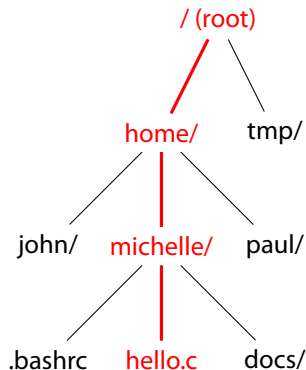
Συστήματα αρχείων

- 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:

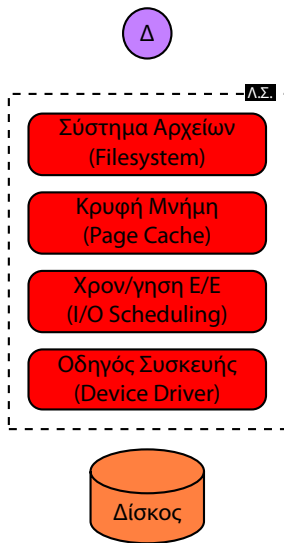
- Άνοιγμα (open)
- Εγγραφή από μνήμη (write)
- Κλείσιμο (close)

Από (εγγραφή) και Προς (ανάγνωση) τη μνήμη

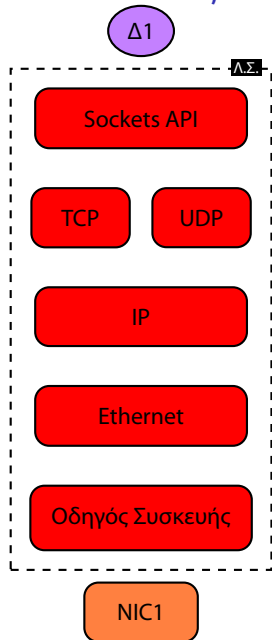
υπάρχουν εξαιρέσεις (πχ `sendfile()`)

ΛΣ και Συσκευές Αποθήκευσης

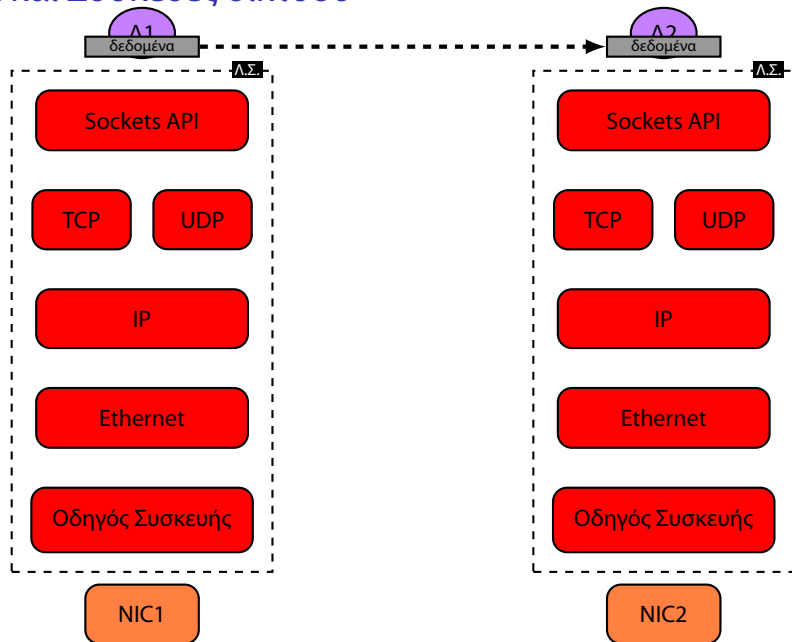
- Συστήματα Αρχείων
 - Ιεραρχική δομή πάνω από γραμμικό χώρο (συσκευή)
- Κρυφή Μνήμη
 - Η πρόσβαση στο δίσκο είναι αργή
 - Περιοχές αρχείων στη μνήμη
- Χρονοδρομολόγηση Ε/Ε
 - Μεγάλος χρόνος αναζήτησης (seek)
 - Βελτιστοποίηση Ε/Ε αιτήσεων



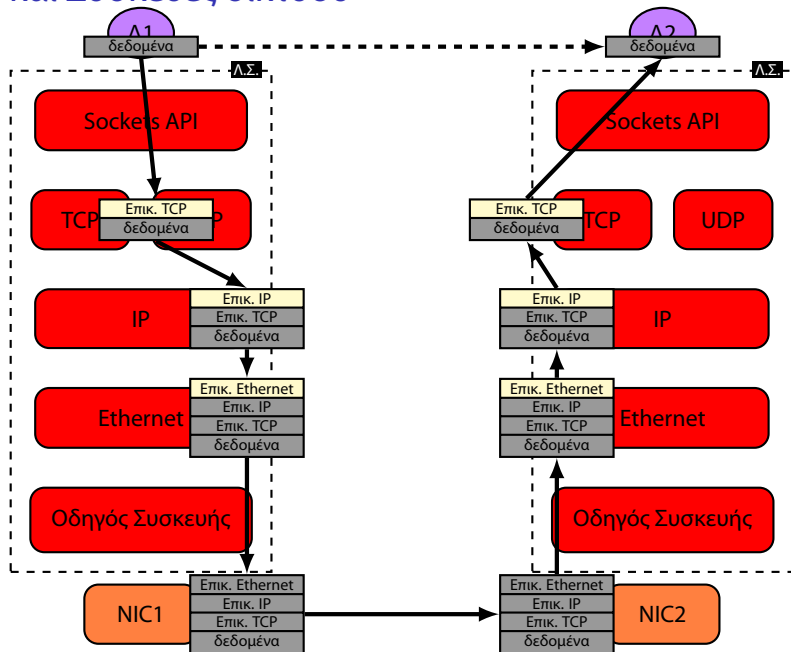
ΛΣ και Συσκευές δικτύου



ΛΣ και Συσκευές δικτύου



ΛΣ και Συσκευές δικτύου



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)
 - Στοιίβα 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

