

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Λειτουργικά Συστήματα

**Ενότητα # 1: Εισαγωγή**

**Διδάσκων: Γεώργιος Ξυλωμένος**

**Τμήμα: Πληροφορικής**



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Οικονομικό Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Οι εικόνες προέρχονται από το βιβλίο «Σύγχρονα Λειτουργικά Συστήματα», A.S. Tanenbaum, 3<sup>η</sup> έκδοση, 2009, Εκδόσεις Κλειδάριθμος.



# Σκοποί ενότητας

- Κατανόηση των στόχων και της ιστορικής εξέλιξης των λειτουργικών συστημάτων (ΛΣ)
- Επισκόπηση του υλικού υπολογιστών
- Κατανόηση των τύπων των ΛΣ
- Παρουσίαση των βασικών εννοιών και των τυπικών κλήσεων των ΛΣ
- Παρουσίαση των τρόπων δόμησης ΛΣ
- Εισαγωγή στη γλώσσα C

# Περιεχόμενα ενότητας

- Τι είναι το λειτουργικό σύστημα (ΛΣ);
- Ιστορία ΛΣ
- Υλικό υπολογιστών
- Είδη ΛΣ
- Έννοιες ΛΣ
- Κλήσεις συστήματος
- Δομή ΛΣ
- Η γλώσσα C

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Τι είναι το λειτουργικό σύστημα;

**Μάθημα:** Λειτουργικά Συστήματα, **Ενότητα # 1:** Εισαγωγή

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

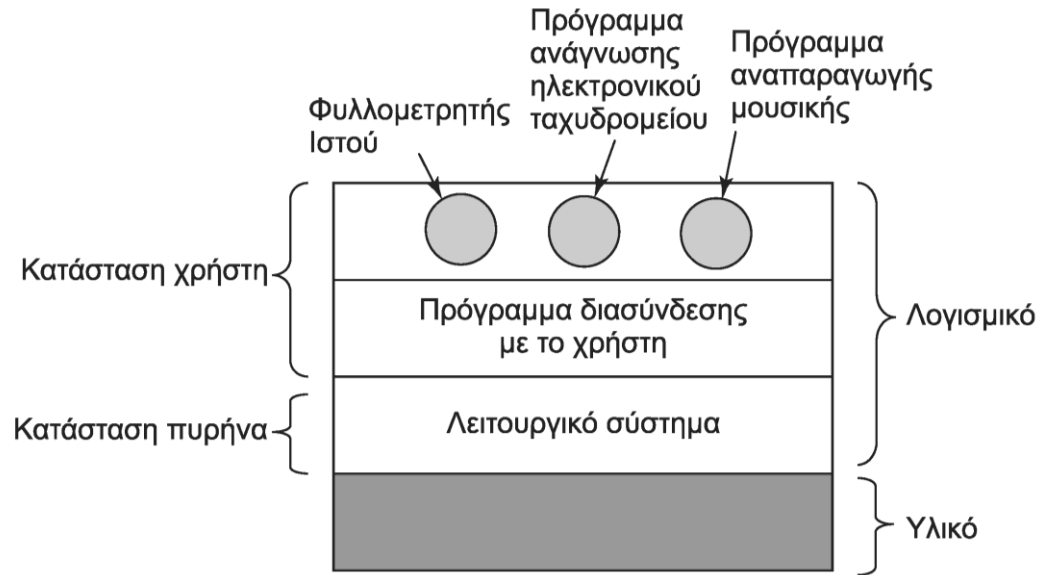
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Τι είναι το ΛΣ (1 από 3);

- Ένας υπολογιστής έχει πολλούς πόρους
  - Επεξεργαστές, μνήμες, δίσκους, εκτυπωτές, ...
- Λειτουργικό σύστημα (ΛΣ)
  - Διαχείριση των πόρων του υπολογιστή
    - Χρονοπρογραμματισμός, έλεγχος πρόσβασης, ...
  - Παροχή διεπαφής προς τον προγραμματιστή
    - Όχι προς τον απλό χρήστη!
  - Το ίδιο ΛΣ μπορεί να έχει πολλές διεπαφές

# Τι είναι το ΛΣ (2 από 3);



- Καταστάσεις λειτουργίας υπολογιστή
  - Κατάσταση πυρήνα ή επόπτη: λειτουργικό σύστημα
  - Κατάσταση χρήστη: διεπαφή χρήστη και εφαρμογές
    - Δεν επιτρέπεται η εκτέλεση ορισμένων εντολών

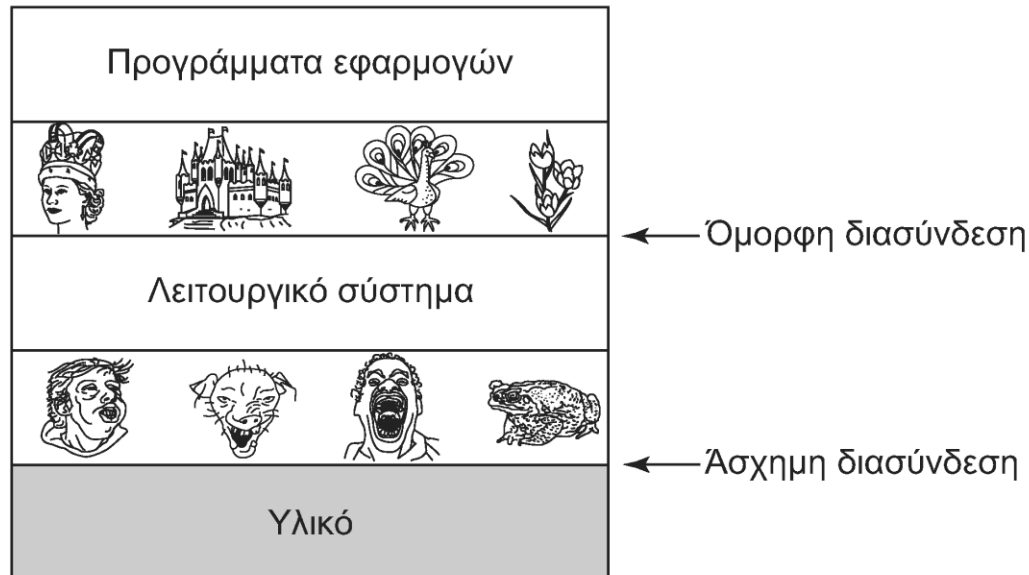


# Τι είναι το ΛΣ (3 από 3);

- Τι ακριβώς περιλαμβάνει το λειτουργικό σύστημα;
  - Βασικό λογισμικό που εκτελείται σε κατάσταση πυρήνα
  - Πιθανόν και προνομιούχο λογισμικό επιπέδου χρήστη
- Γιατί μελετάμε τα λειτουργικά συστήματα χωριστά;
  - Πολύ πιο περίπλοκα από τα προγράμματα εφαρμογών
  - Δύσκολο να γραφτούν νέα λόγω πολυπλοκότητας
- Κύρια λειτουργικά συστήματα
  - UNIX και απόγονοι: Linux, xBSDy, OSX
  - MS-DOS και απόγονοι έως Windows ME
  - Windows NT και απόγονοι έως Windows 8

# Η επεκτεταμένη μηχανή (1 από 2)

- Όψη του λειτουργικού συστήματος από πάνω
  - Μετατροπή μιας άσχημης εικόνας σε πιο όμορφη
- Το υλικό των υπολογιστών είναι περίπλοκο
  - Κάθε συσκευή έχει τις δικές τις ιδιαιτερότητες



# Η επεκτεταμένη μηχανή (2 από 2)

- Το λειτουργικό σύστημα παρέχει λογικές αφαιρέσεις
  - Κάθε είδος δίσκου είναι πολύ διαφορετικό
  - Το λειτουργικό σύστημα μας παρουσιάζει λογικά αρχεία
  - Μετατροπή εντολών αρχείων σε εντολές προς το δίσκο
- Σε ποιον παρέχεται η λογική αφαίρεση;
  - Κυρίως στους προγραμματιστές των εφαρμογών
    - Πρόκειται για διεπαφή προγραμματισμού
  - Οι χρήστες συνήθως βλέπουν τη διεπαφή χρήστη
    - Πρόκειται για γραμμή εντολών ή γραφικό περιβάλλον

# Ο διαχειριστής πόρων

- Όψη του λειτουργικού συστήματος από κάτω
  - Διαχείριση των διάφορων πόρων ενός συστήματος
  - Μας ενδιαφέρει η αποδοτική διαχείριση των πόρων
- Δυνατότητα ταυτόχρονης εκτέλεσης προγραμμάτων
  - Κάθε πρόγραμμα θεωρεί ότι έχει τη δική του μηχανή
- Διαχείριση και προστασία μνήμης και συσκευών
- Πολύπλεξη πόρων σε δύο άξονες
  - Χρόνος: διαδοχική χρήση ΚΜΕ από τα προγράμματα
  - Χώρος: συνύπαρξη προγραμμάτων στη μνήμη

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**

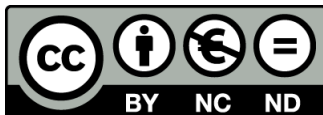


**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Ιστορία ΛΣ

**Μάθημα:** Λειτουργικά Συστήματα, **Ενότητα # 1:** Εισαγωγή

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Ιστορία ΛΣ (1 από 10)

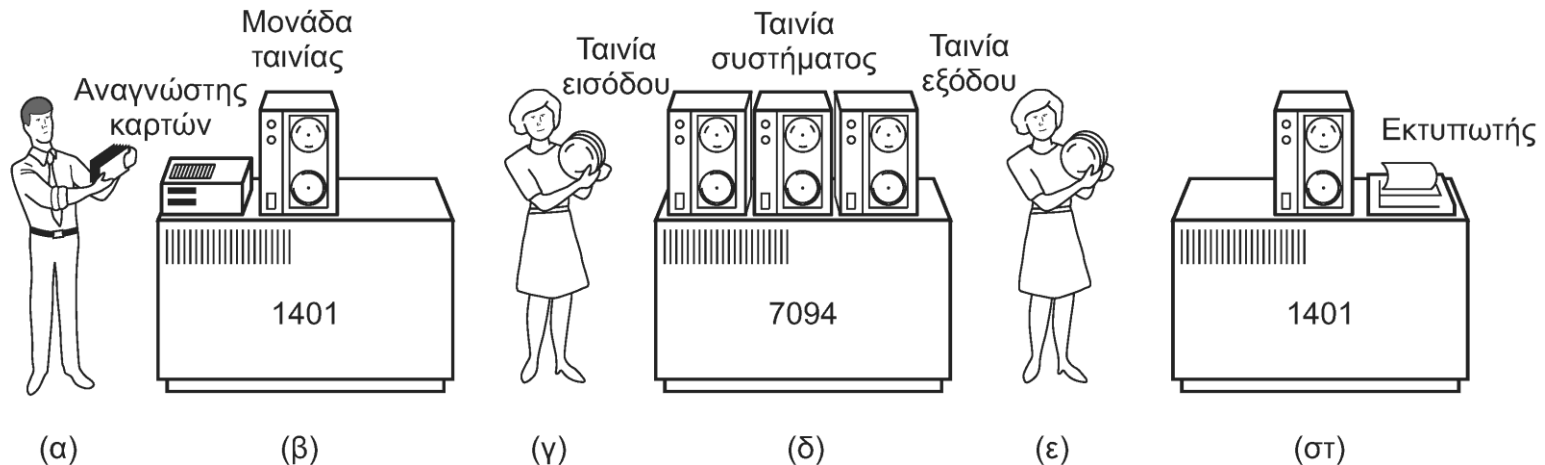
- Πρώτη γενιά (1945-1955): λυχνίες κενού
  - Παράδειγμα: ENIAC, UNIVAC
  - Προγραμματισμός από τους σχεδιαστές
  - Απόλυτη γλώσσα μηχανής ή πίνακες καλωδιώσεων
  - Σχετικά απλές μηχανές με απλές λειτουργίες
  - Δεν υπήρχε λειτουργικό σύστημα
- Χρήση διάτρητων καρτών για προγραμματισμό
  - Αρχικά χρήση για είσοδο δεδομένων
  - Αντικατέστησε τους πίνακες καλωδιώσεων

# Ιστορία ΛΣ (2 από 10)

- Δεύτερη γενιά (1955-1965): τρανζίστορ
  - Μηχανές αρκετά αξιόπιστες έτσι ώστε να πωλούνται
  - Οι προγραμματιστές δεν είναι πια οι σχεδιαστές
  - Μεγάλα υπολογιστικά συστήματα (mainframes)
- Υποβολή και εκτέλεση εργασιών (jobs)
  - Κάθε εργασία αποτελείται από μια σειρά κάρτες
  - Ο προγραμματιστής δίνει τις κάρτες στο χειριστή
  - Ο χειριστής υποβάλει τις εργασίες με τη σειρά
  - Σημαντικός χρόνος αδράνειας του υπολογιστή

# Ιστορία ΛΣ (3 από 10)

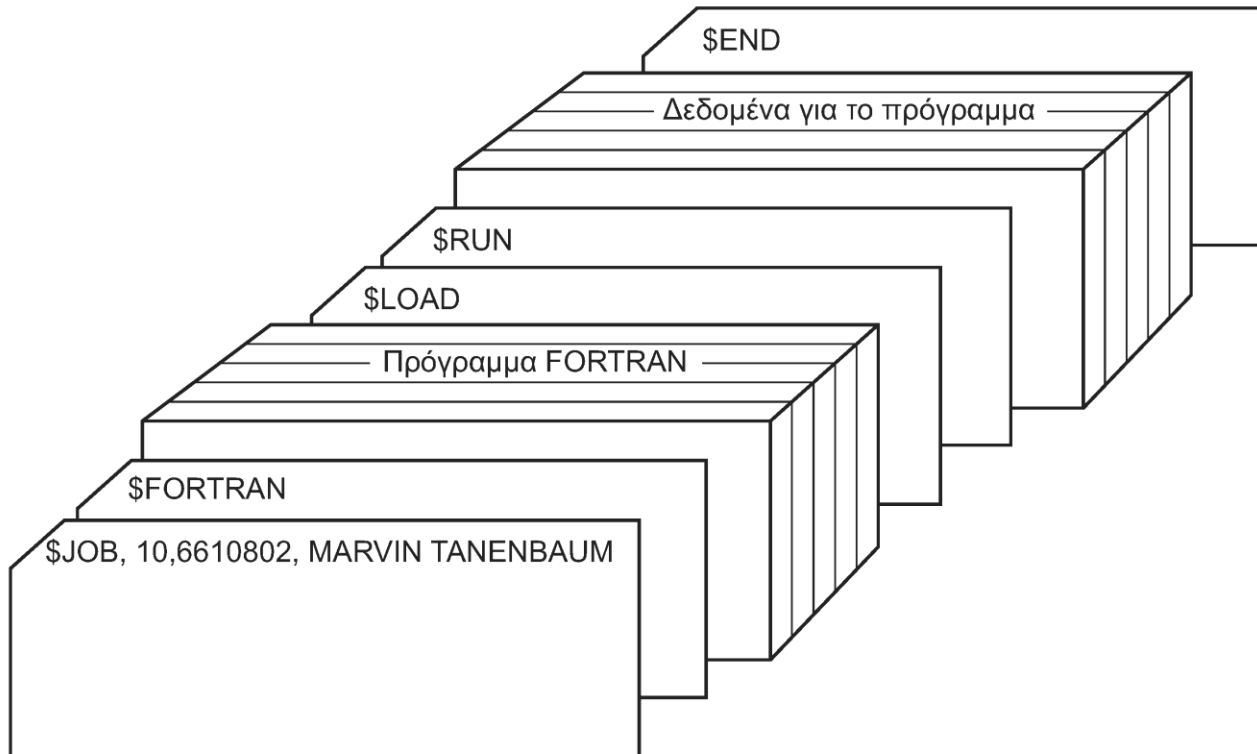
- Αυτοματοποίηση συστημάτων δέσμης
  - Αντιγραφή εργασιών σε ταινίες σε μικρό υπολογιστή
  - Υποβολή ταινίας με εργασίες στον μεγάλο υπολογιστή
  - Έξοδος αποτελεσμάτων σε άλλη ταινία
  - Εκτύπωση αποτελεσμάτων σε μικρό υπολογιστή





# Ιστορία ΛΣ (4 από 10)

- Λειτουργικά συστήματα δέσμης
  - Ειδικές κάρτες ελέγχου μιας εργασίας (\$)
  - Κάρτες προγράμματος και δεδομένων

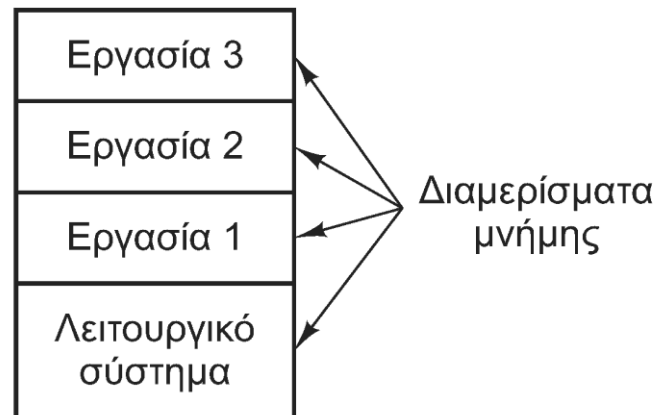


# Ιστορία ΛΣ (5 από 10)

- Τρίτη γενιά (1965-1980): ολοκληρωμένα κυκλώματα
  - Αντικατάσταση τρανζίστορ με κυκλώματα SSI
  - Προσανατολισμός σε συγκεκριμένες εφαρμογές
- Οικογένεια υπολογιστών
  - Ιδέα της IBM με το System/360
  - Μηχανές με ίδια αρχιτεκτονική και σύνολο εντολών
  - Διαφορετικές δυνατότητες και κόστος
  - Ίδια προγράμματα (εντός λογικών πλαισίων)
  - Χρειάζεται και το ίδιο λειτουργικό σύστημα
  - Εξαιρετικά περίπλοκο λόγω των διαφορετικών μηχανών

# Ιστορία ΛΣ (6 από 10)

- Πολυπρογραμματισμός
  - Ψευδοταυτόχρονη λειτουργία πολλών εργασιών
  - Κάθε εργασία βρίσκεται σε χωριστό μέρος τη μνήμης
  - Όταν μια εργασία περιμένει, εκτελείται κάποια άλλη
  - Ταυτόχρονη επεξεργασία και είσοδος/έξοδος (spooling)
  - Χρειάζεται ειδικό υλικό προστασίας των εργασιών



# Ιστορία ΛΣ (7 από 10)

- Χρονομερισμός
  - Βασίζεται στην πολυπρογραμματισμό
  - Κάθε χρήστης έχει το δικό του τερματικό
  - Το σύστημα εξυπηρετεί τους χρήστες ψευδοταυτόχρονα
- Το σύστημα MULTICS
  - Παροχή υπολογιστικής ισχύος από κεντρικό υπολογιστή
  - Όπως το cloud computing αλλά με έναν υπολογιστή!
- Τα συστήματα DEC PDP
  - Μικροί και (σχετικά οικονομικοί) υπολογιστές
  - Μεγάλη εξάπλωση σε περισσότερους οργανισμούς

# Ιστορία ΛΣ (8 από 10)

- MULTICS+PDP = UNIX
  - Επηρέασε όλα τα μεταγενέστερα συστήματα
  - Το UNIX εξελίχθηκε σε System V και BSD
  - Το MINIX γράφτηκε για εκπαιδευτικούς σκοπούς
    - Μικροπυρήνας και έμφαση στην απλότητα
    - Στόχος: ένα UNIX που να το καταλαβαίνουν οι φοιτητές
  - Το Linux γράφτηκε ως πρακτική έκδοση του MINIX
    - Μονολιθικός πυρήνας και έμφαση στη λειτουργικότητα
    - Στόχος: ένα UNIX που να κάνει οποιαδήποτε δουλειά

# Ιστορία ΛΣ (9 από 10)

- Τέταρτη γενιά (1980-): προσωπικοί υπολογιστές
  - Χρήση κυκλωμάτων VLSI με τεράστιες δυνατότητες
  - Δυνατότητα τοποθέτησης ΚΜΕ σε ένα κύκλωμα
  - Εμφάνιση προσωπικών υπολογιστών
- ΛΣ προσωπικών υπολογιστών
  - Το CP/M γράφτηκε για τον 8080 και μετά τον Z80
  - Το MS-DOS γράφτηκε για τον 8088/8086
    - Αρχικά σαν το CP/M, μετά με ιδέες από το UNIX
  - Εμφάνιση υπολογιστών με γραφικές διεπαφές

# Ιστορία ΛΣ (10 από 10)

- Τα παλιά Microsoft Windows
  - Γραφική διεπαφή πάνω από το MS-DOS
  - Windows 3, 95, 98, Me
- Τα νέα Microsoft Windows
  - Υλοποίηση από την αρχή ξεκινώντας από τα Windows NT
  - Windows 2000, XP, 7
- Τα συστήματα UNIX
  - Προσθήκη γραφικής διεπαφής: X Window System
  - Πολύ μεγάλη εξάπλωση του Linux
  - Το UNIX (τύπου BSD) αντικατέστησε και το αρχικό Mac OS

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Υλικό υπολογιστών

**Μάθημα:** Λειτουργικά Συστήματα, **Ενότητα # 1:** Εισαγωγή

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

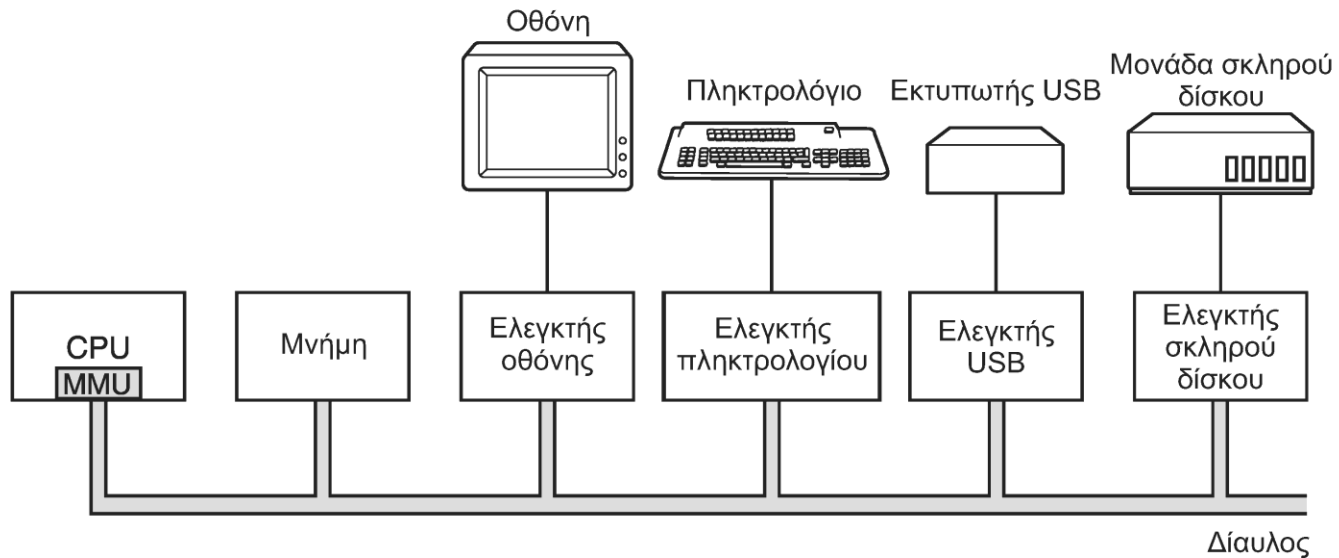
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



# ΛΣ και υλικό

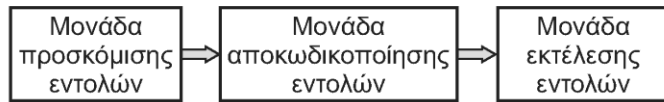


- Λειτουργικό σύστημα και υλικό
  - Το λειτουργικό σύστημα παρέχει εκτεταμένη μηχανή
  - Πρέπει όμως να κατανοεί την πραγματική μηχανή
  - Βασική δομή υπολογιστικού συστήματος

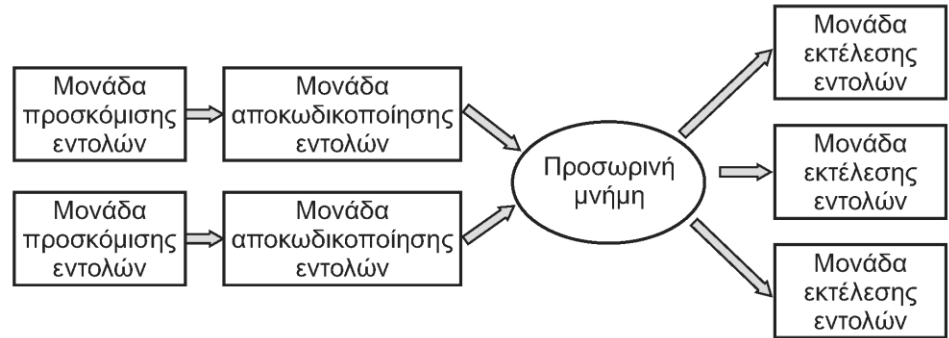
# Επεξεργαστής (1 από 4)

- Επεξεργαστής
  - Βασικός κύκλος λειτουργίας
    - Προσκόμιση, αποκωδικοποίηση, τελεστές, εκτέλεση
  - Καταχωρητές δεδομένων
    - Αποθήκευση δεδομένων για γρήγορη πρόσβαση
  - Καταχωρητές ελέγχου
    - Μετρητής προγράμματος, δείκτης στοίβας, κατάσταση
  - Αποθήκευση καταχωρητών κατά την εναλλαγή
    - Λειτουργία με αρκετά μεγάλο κόστος

# Επεξεργαστής (2 από 4)



(α)



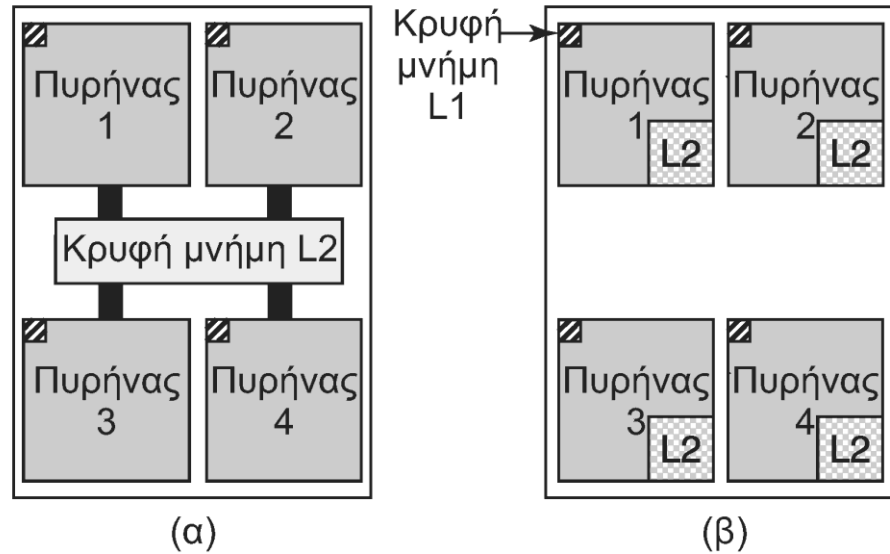
(β)

- Σωλήνωση
  - Χωριστές μονάδες για κάθε στάδιο εκτέλεσης
  - Εκτέλεση διαφορετικής εντολής σε κάθε ομάδα
- Υπερβαθμωτοί επεξεργαστές
  - Πολλαπλές μονάδες εκτέλεσης
  - Προσκόμιση και αποκωδικοποίηση πολλών εντολών
  - Εκτέλεση εντολών και εκτός σειράς

# Επεξεργαστής (3 από 4)

- Καταστάσεις λειτουργίας επεξεργαστή
  - Κατάσταση χρήστη και κατάσταση πυρήνα
  - Σε κατάσταση χρήστη επιτρέπεται υποσύνολο εντολών
- Κλήσεις συστήματος
  - Εντολές που προκαλούν παγίδα υλικού
  - Μετάβαση σε κατάσταση πυρήνα με ελεγχόμενο τρόπο
- Πολυνηματικοί και πολυπύρρηνοι επεξεργαστές
  - Πολλοί τρόποι αξιοποίησης διαθέσιμων τρανζίστορ
  - Κρυφή μνήμη, πολλαπλές μονάδες εκτέλεσης
  - Γιατί όχι και πολλές μονάδες ελέγχου;

# Επεξεργαστής (4 από 4)



- Πολυνηματικοί και πολυπύρρηνοι επεξεργαστές
  - Πολυνημάτωση: εκτέλεση πολλών νημάτων
  - Πολυπύρρηνος επεξεργαστής: ανεξάρτητες CPU
    - Κανονική πολυεπεξεργασία σε ένα ολοκληρωμένο

# Μνήμη (1 από 2)

Τυπικός χρόνος προσπέλασης

Τυπική αποθηκευτική ικανότητα

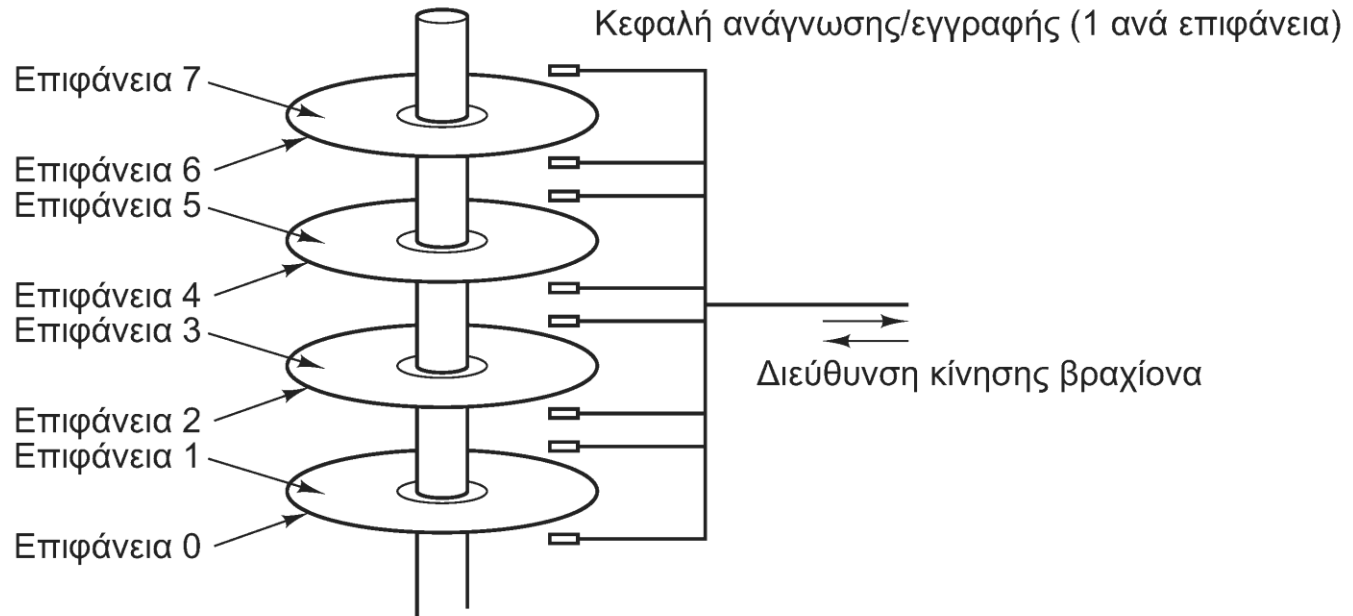


- Μνήμη και ιεραρχία μνήμης
  - Κάθε τύπος μνήμης παρέχει έναν συμβιβασμό
  - Ταχύτητα έναντι κόστους (άρα και μεγέθους)
  - Μνήμη ROM (ή EEPROM και Flash) για βασικό λογισμικό
  - Μνήμη CMOS για διάρθρωση συστήματος

# Μνήμη (2 από 2)

- Κρυφές μνήμες
  - Οργανώνονται και αυτές ιεραρχικά
  - Γενικά θέματα οργάνωσης κρυφών μνημών
  - Πολιτικές προσκόμισης, τοποθέτησης, αντικατάστασης
- Κρυφή μνήμη επιπέδου 1 (L1)
  - Στο εσωτερικό του επεξεργαστή
  - Συνήθως χωριστή για εντολές και δεδομένα
- Κρυφή μνήμη επιπέδου 2 (L2)
  - Εντός ή εκτός επεξεργαστή
  - Κοινή ή καταμεριζόμενη ανάμεσα στους πυρήνες

# Αποθήκευση



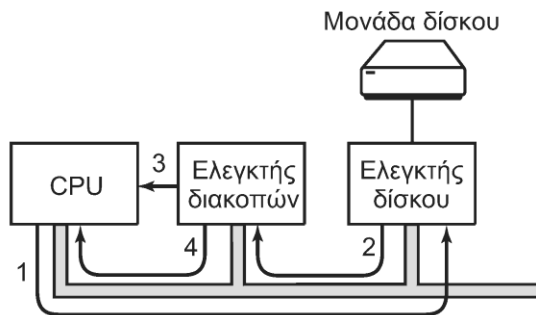
- Μαγνητικοί δίσκοι
  - Τροχιές/κύλινδροι, επιφάνειες, τομείς
  - Αρκετός χρόνος μέχρι να βρεθεί το σωστό σημείο
  - Αποθήκευση δεδομένων και προγραμμάτων



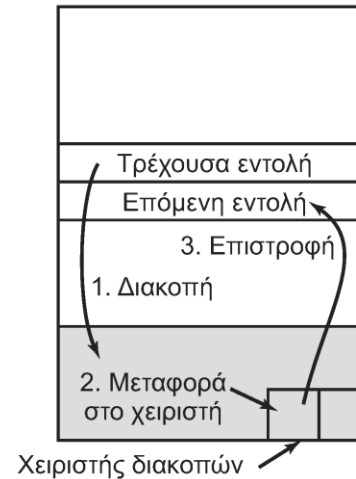
# Είσοδος / Έξοδος (1 από 2)

- Συσκευές εισόδου/εξόδου
  - Ελεγκτής και πραγματική συσκευή
  - Ο ελεγκτής παρουσιάζει απλούστερη διεπαφή
  - Οδηγός συσκευής: λογισμικό για τον ελεγκτή
  - Επικοινωνία με τους ελεγκτές των συσκευών
    - Απεικόνιση καταχωρητών στη μνήμη
    - Χωριστές εντολές εισόδου/εξόδου
  - Έλεγχος συσκευών: polling, interrupts, DMA

# Είσοδος / Έξοδος (2 από 2)



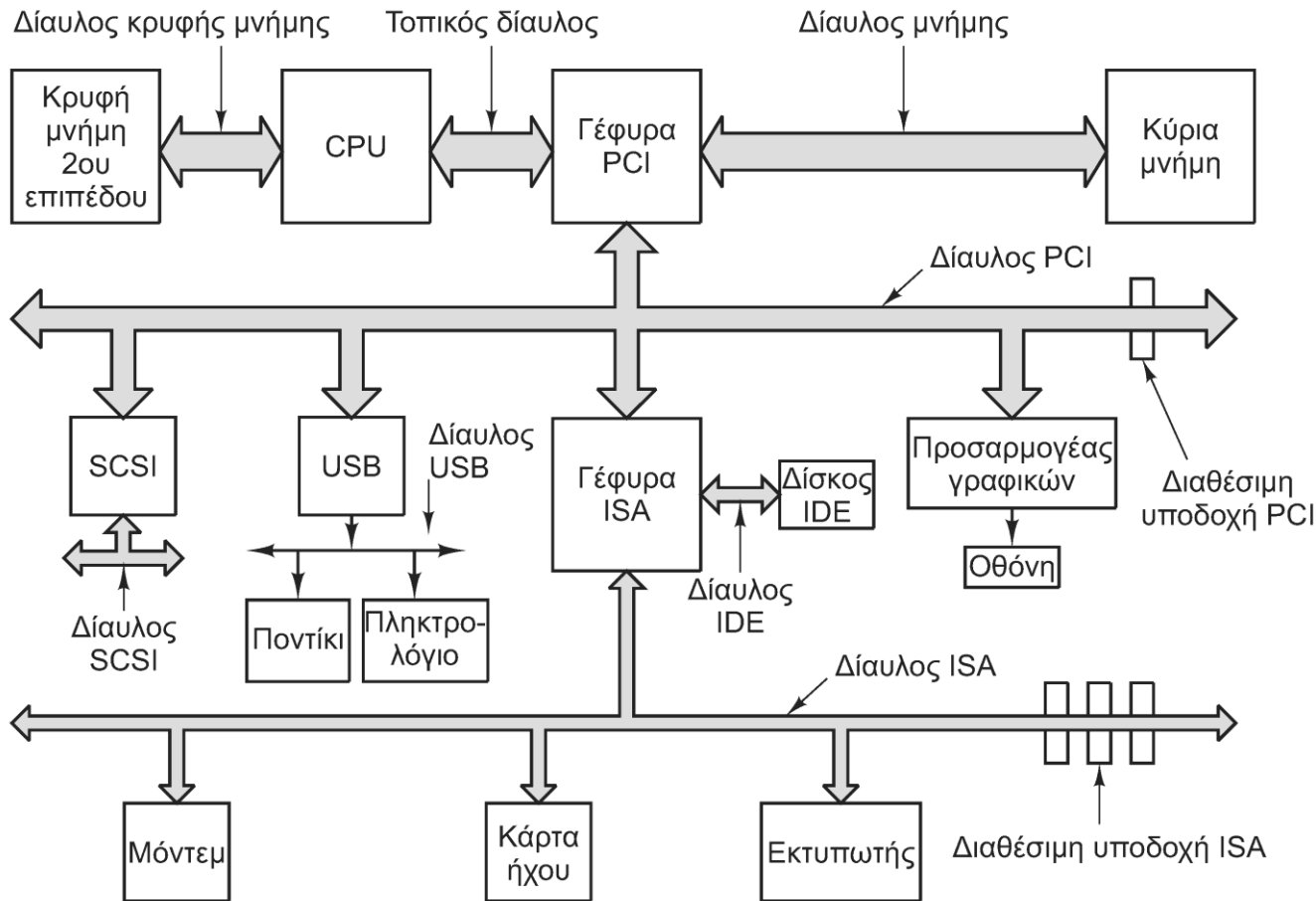
(α)



(β)

- Είσοδος/έξοδος με διακοπές
  - Ο ελεγκτής διακόπτει την CPU όταν ολοκληρώσει
  - Μεταφορά ελέγχου στο κατάλληλο σημείο της μνήμης
- Είσοδος/έξοδος με DMA
  - Ο ελεγκτής μεταφέρει ο ίδιος τα δεδομένα στη μνήμη

# Δίαυλοι (1 από 2)



- Λεωφόροι (δίαυλοι) συστήματος

# Δίαυλοι (2 από 2)

- Δίαυλοι συστήματος
  - Επικοινωνία με κρυφή μνήμη με ειδικό δίαυλο
  - Τοπικός δίαυλος για επικοινωνία με το σύστημα
  - Γέφυρα PCI για επικοινωνία με μνήμη ή συσκευές
  - Δίαυλος PCI/PCI Express για τις περισσότερες συσκευές
  - Γέφυρα ISA για χρήση διαύλου ISA (παλιές συσκευές)
  - Δίαυλος USB για (αργές) σειριακές συσκευές
  - Δίαυλος SCSI για (γρήγορους) δίσκους
  - Δίαυλος SATA για πιο οικονομικούς δίσκους
  - Τοποθέτηση και άμεση λειτουργία (plug and play)

# Εκκίνηση ΛΣ

- Βασικό λογισμικό συστήματος στο BIOS
  - Εξετάζει τις βασικές συσκευές σε γνωστές θέσεις
  - Εξετάζει τους διαύλους για πρόσθετες συσκευές
  - Καθορισμός συσκευής εκκίνησης μέσω μνήμης CMOS
  - Φόρτωση τομέα 0 από συσκευή εκκίνησης
  - Προσδιορισμός ενεργής διαμέρισης
  - Φόρτωση προγράμματος εκκίνησης από τη διαμέριση
  - Φόρτωση λειτουργικού συστήματος
  - Εξέταση συσκευών από BIOS και φόρτωση οδηγών
  - Αρχικοποίηση δομών δεδομένων και φόρτωση φλοιού

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Είδη ΛΣ

**Μάθημα:** Λειτουργικά Συστήματα, **Ενότητα # 1:** Εισαγωγή

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# Είδη ΛΣ (1 από 3)

- ΛΣ για μεγάλους υπολογιστές (mainframes)
  - Πολύ μεγάλες ικανότητες εισόδου/εξόδου
  - Επεξεργασία δέσμης, συναλλαγών και χρονομερισμός
- ΛΣ για διακομιστές (servers)
  - Διαχείριση αιτημάτων πάρα πολλών πελατών
  - Μεγάλοι υπολογιστές ή ισχυροί μικροϋπολογιστές
- ΛΣ για πολυεπεξεργαστές (multiprocessors)
  - Από παράλληλοι ως πολυπύρηντοι υπολογιστές
  - Συνήθως παραλλαγές των ΛΣ για διακομιστές

# Είδη ΛΣ (2 από 3)

- ΛΣ προσωπικών υπολογιστών (personal computers)
  - Τα συνηθισμένα συστήματα ενός (κυρίως) χρήστη
- ΛΣ για υπολογιστές χειρός (handhelds)
  - Μικρότερη δυνατότητα αποθήκευσης (χωρίς δίσκους)
- Ενσωματωμένα ΛΣ (embedded)
  - Ελέγχουν συσκευές όπως DVD, τηλέφωνα, media players
  - Εκτελούν μόνο προκαθορισμένα προγράμματα
- ΛΣ κόμβων αισθητήρων (sensors)
  - Πολύ απλά συστήματα οδηγούμενα από γεγονότα



# Είδη ΛΣ (3 από 3)

- ΛΣ πραγματικού χρόνου
  - Αυστηρά συστήματα πραγματικού χρόνου
    - Όλες οι εργασίες πρέπει να εκτελούνται πάντα εγκαίρως
    - Συστήματα ελέγχου, ηλεκτρονικά αεροσκαφών
  - Ήπια συστήματα πραγματικού χρόνου
    - Οι εργασίες πρέπει να εκτελούνται συνήθως εγκαίρως
    - Κινητά τηλέφωνα, media players
- ΛΣ έξυπνων καρτών
  - Εκτελούνται σε έξυπνες κάρτες με ελάχιστους πόρους

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Έννοιες ΛΣ

**Μάθημα:** Λειτουργικά Συστήματα, **Ενότητα # 1:** Εισαγωγή

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

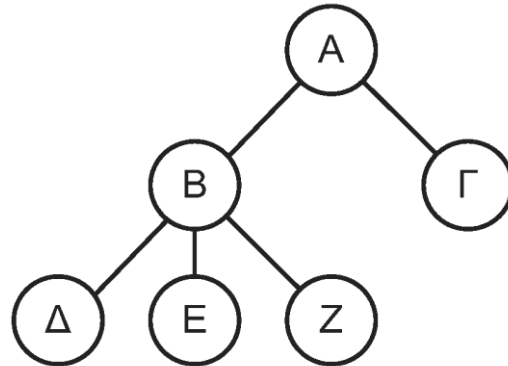


ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# Διεργασίες (1 από 2)

- Διεργασίες: προγράμματα που εκτελούνται
  - Χώρος διευθύνσεων: πρόγραμμα και δεδομένα
  - Πόροι: καταχωρητές, ανοιχτά αρχεία, σήματα
- Προσωρινή αναστολή διεργασίας
  - Αποθήκευση πόρων στον πίνακα διεργασιών
  - Ο πίνακας διεργασιών είναι συνέχεια στη μνήμη
  - Ο χώρος διευθύνσεων μπορεί να είναι εν μέρει στη μνήμη
- Κλήσεις διαχείρισης διεργασιών
  - Δημιουργία και τερματισμός διεργασιών
  - Επικοινωνία διεργασιών (όταν αυτό επιτρέπεται)

# Διεργασίες (2 από 2)



- Δένδρα διεργασιών
- Σήματα (signals)
  - Ειδοποιήσεις προς τη διεργασία για κάποιο γεγονός
  - Παράδειγμα: εκπνοή χρονομέτρου (alarm)
- Προνόμια διεργασίας
  - Κωδικός ταυτότητας χρήστη (UID)
  - Κωδικός ταυτότητας ομάδας (GID)

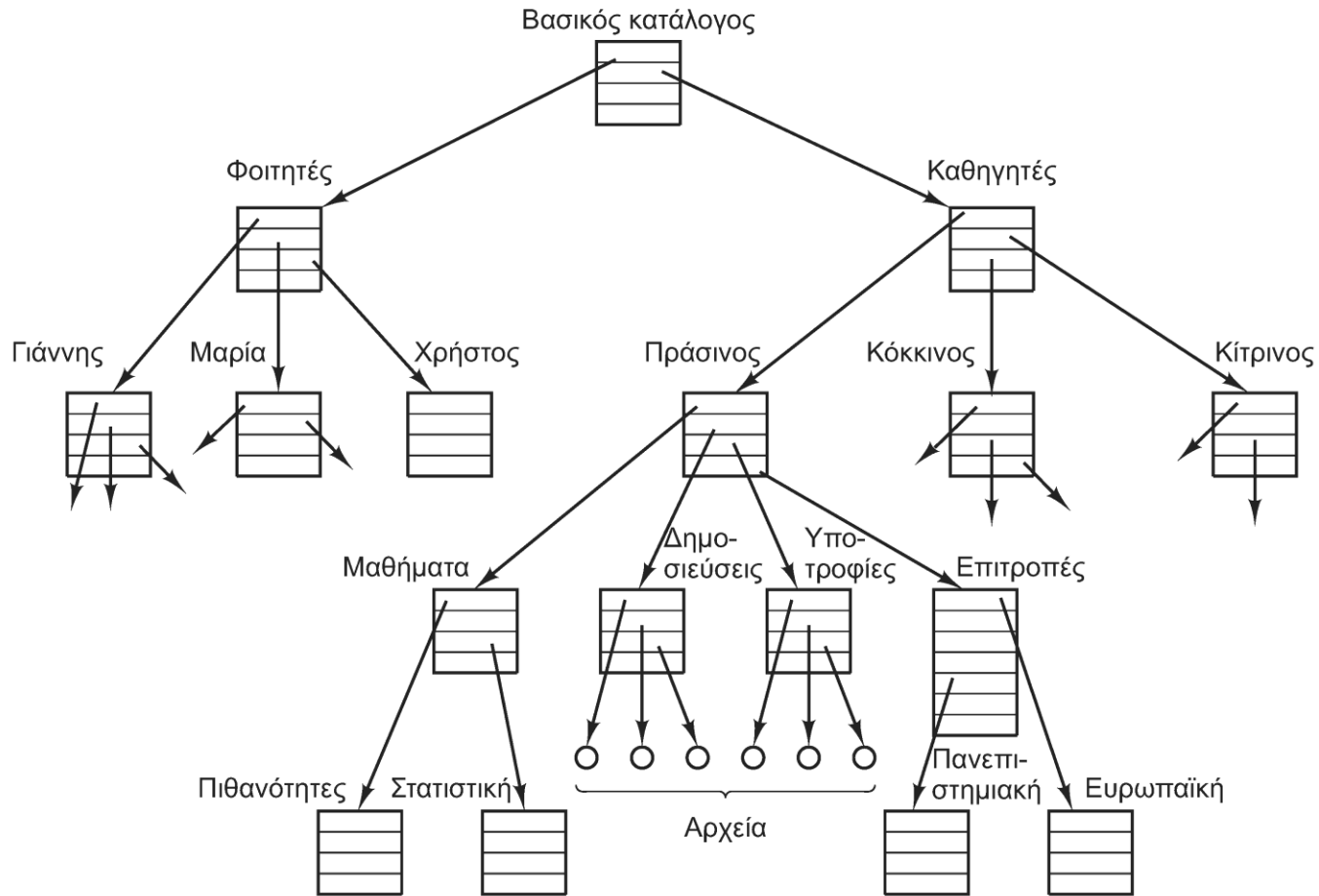
# Χώροι διευθύνσεων

- Χώροι διευθύνσεων
  - Τα ΛΣ γενικά υποστηρίζουν πολλές διεργασίες
  - Μηχανισμός προστασίας κάθε διεργασίας
    - Απομόνωση της μνήμης της
  - Εικονική μνήμη: μεγάλος χώρος διευθύνσεων
    - Δεν είναι όλος ταυτόχρονα στην κύρια μνήμη
    - Διευκολύνει τη συνύπαρξη διεργασιών στη μνήμη
    - Ό,τι δεν είναι στην κύρια μνήμη είναι στο δίσκο

# Αρχεία (1 από 4)

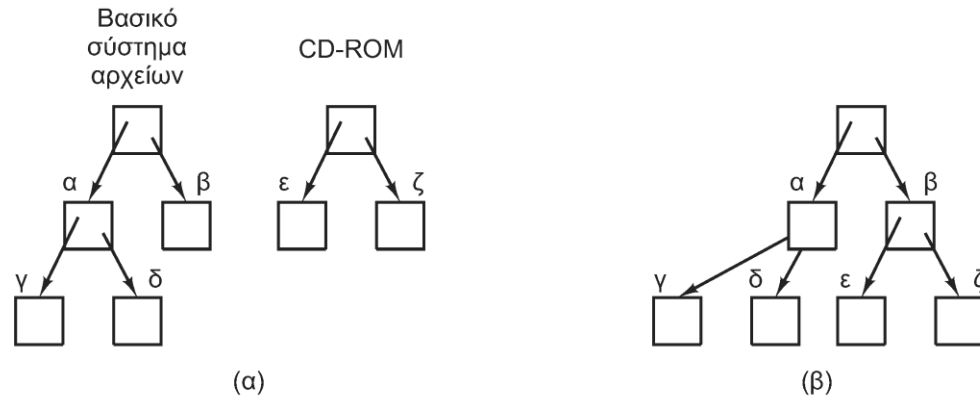
- Αρχεία
  - Λογική όψη συσκευών αποθήκευσης
    - Ενιαία μορφή αρχείων σε όλες τις συσκευές
  - Οργάνωση αρχείων σε καταλόγους
  - Περιγραφέας αρχείου: δείκτης σε ανοιχτό αρχείο
    - Χρησιμοποιείται από τις διεργασίες
    - Χειριστήριο για πράξεις στα αρχεία

# Αρχεία (2 από 4)



- Παράδειγμα δενδρικής οργάνωσης αρχείων

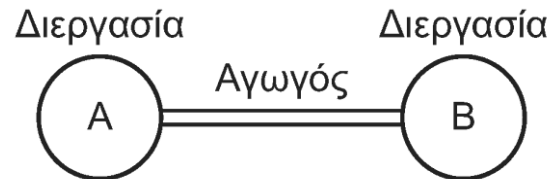
# Αρχεία (3 από 4)



- Όνομα διαδρομής αρχείου
  - Απόλυτο: ξεκινάει από τη ρίζα
  - Σχετικό: ξεκινάει από τον κατάλογο εργασίας
- Ανάρτηση συστημάτων αρχείων
  - Εκκίνηση με το βασικό σύστημα αρχείων
  - Ανάρτηση πρόσθετων συστημάτων ανάλογα με τις ανάγκες



# Αρχεία (4 από 4)



- Ειδικά αρχεία
  - Απεικόνιση των συσκευών σε αρχεία
  - Ειδικά αρχεία μπλοκ ή ομάδων
    - Συσκευές με τυχαία προσπελάσιμα μπλοκ (π.χ. δίσκοι)
  - Ειδικά αρχεία χαρακτήρων
    - Συσκευές με ρεύματα χαρακτήρων (π.χ. πληκτρολόγια)
- Αγωγοί ή σωληνώσεις (pipes)
  - Ψευδοαρχείο ανάμεσα σε δύο διεργασίες
  - Επιτρέπει την επικοινωνία των διεργασιών

# Άλλες έννοιες

- Είσοδος/έξοδος
  - Υποσύστημα διαχείρισης συσκευών εισόδου/εξόδου
  - Οδηγοί συσκευών ανά κατηγορία συσκευής
- Προστασία
  - Μηχανισμός που προστατεύει τα στοιχεία των διεργασιών
  - Παράδειγμα: bit προστασίας στο UNIX
- Κέλυφος
  - Δεν είναι μέρος του λειτουργικού (αλλάζει «εύκολα»)
  - Επιτρέπει στον χρήστη να επικοινωνεί με το λειτουργικό
  - Γραμμή εντολών ή γραφική διεπαφή

# Ανακύκλωση ιδεών (1 από 3)

- Υπολογιστές και ΛΣ ανακυκλώνουν ιδέες
  - Οι αλλαγές στην τεχνολογία οδηγούν σε νέες ιδέες
  - Οι παλιές ιδέες επανέρχονται μετά από άλλες αλλαγές
  - Δεν υπάρχουν απαρχαιωμένες, μόνο μη επίκαιρες ιδέες
- Υλοποίηση CPU
  - Αρχικά καλωδιωμένη (πιο απλή)
  - Στη συνέχεια μικροπρογραμματιζόμενη (πιο ευέλικτη)
  - Μετά πάλι καλωδιωμένη (πιο αποδοτική)
  - Η Java μοιάζει με την μικροπρογραμματιζόμενη εκτέλεση

# Ανακύκλωση ιδεών (2 από 3)

- Μεγάλες μνήμες
  - Οι μεγάλοι υπολογιστές ξεκίνησαν με πολύ μικρές μνήμες
  - Όλα τα προγράμματα σε συμβολική γλώσσα
  - Η αύξηση μνήμης οδήγησε στους μεταγλωττιστές
  - Οι μίνι/μίκρο-υπολογιστές γύρισαν σε συμβολική γλώσσα
- Υλικό προστασίας
  - Οι μεγάλοι υπολογιστές δεν είχαν υλικό προστασίας
  - Τα ΛΣ δεν είχαν πολυπρογραμματισμό
  - Το υλικό προστασίας επέτρεψε πολυπρογραμματισμό
  - Το ίδιο έγινε με μίνι/μίκρο-υπολογιστές

# Ανακύκλωση ιδεών (3 από 3)

- Δίσκοι
  - Οι δίσκοι έφεραν τα συστήματα αρχείων
  - Αρχικά ενιαίος κατάλογος για όλο το σύστημα
  - Στη συνέχεια ένας κατάλογος ανά χρήστη
  - Τελικά δενδρικές και άλλες δομές
  - Ίδια ακριβώς εξέλιξη σε μίνι/μικρο-υπολογιστές
- Εικονική μνήμη
  - Έγινε δυνατή με ειδικό υλικό απεικόνισης και προστασίας
  - Αρχικά μόνο σε μεγάλους υπολογιστές
  - Στη συνέχεια σε μίνι/μικρο-υπολογιστές

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**

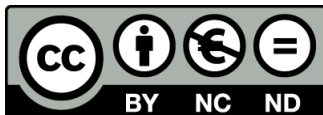


**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Κλήσεις συστήματος

**Μάθημα:** Λειτουργικά Συστήματα, **Ενότητα # 1:** Εισαγωγή

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

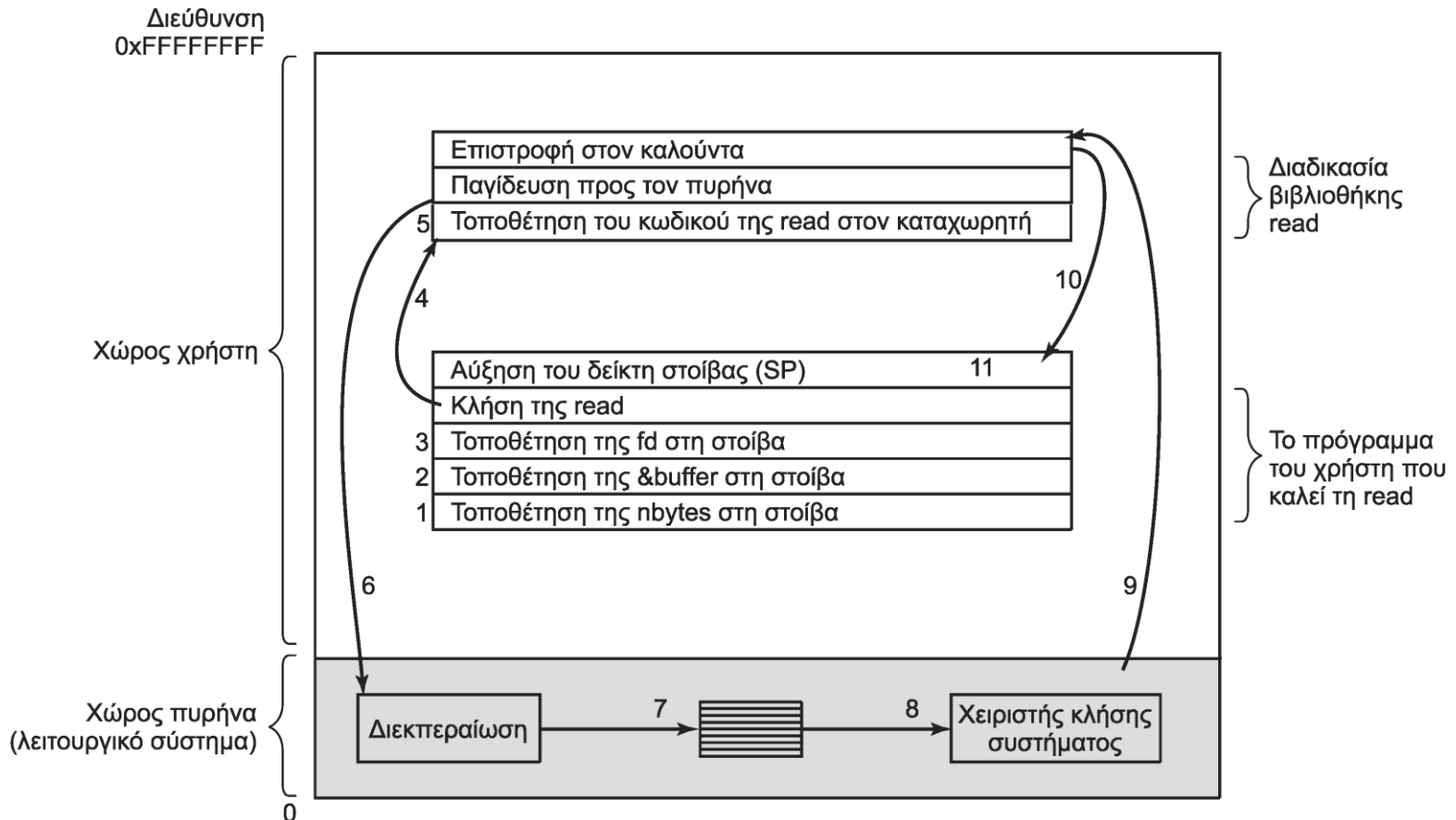
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Κλήσεις συστήματος (1 από 3)

- Οι δύο σκοποί των λειτουργικών συστημάτων
  - Διαχείριση πόρων: αόρατη στις διεργασίες
  - Παροχή αφαιρέσεων: (πολύ) ορατή στα προγράμματα
- Διασύνδεση προγραμμάτων με το ΛΣ
  - Χρήση του POSIX (UNIX) ως παράδειγμα
  - Περιγράφεται με μία σειρά κλήσεων βιβλιοθήκης
    - Τα δύο είδη κλήσεων δεν έχουν απόλυτη αντιστοιχία
  - Οι κλήσεις αυτές ενθυλακώνουν τις κλήσεις συστήματος
    - Μετάβαση από κατάσταση χρήστη σε κατάσταση συστήματος
    - Εκτέλεση εργασίας και επιστροφή σε κατάσταση χρήστη

# Κλήσεις συστήματος (2 από 3)



- Παράδειγμα: `count=read(fd,&buffer,nbytes);`



# Κλήσεις συστήματος (3 από 3)

- Βήματα 1-3: τοποθέτηση παραμέτρων στη στοίβα
- Βήμα 4: κλήση διαδικασίας read
- Βήμα 5: προετοιμασία παραμέτρων κλήσης
- Βήμα 6: χρήση εντολής TRAP για την κλήση
- Βήμα 7: μετάβαση σε χειριστή κλήσης
- Βήμα 8: εκτέλεση ζητούμενης εργασίας
- Βήμα 9: επιστροφή στην εντολή μετά την TRAP
- Βήμα 10: επιστροφή στον καλούντα
- Βήμα 11: καθάρισμα στοίβας

# Διαχείριση διεργασιών (1 από 3)

Κλήση	Περιγραφή δράσης
<code>pid = fork()</code>	Δημιουργεί μια θυγατρική διεργασία, πανομοιότυπη με τη γονική
<code>pid = waitpid(pid, &amp;statloc, επιλογές)</code>	Οδηγεί σε αναμονή μέχρι τον τερματισμό της θυγατρικής διεργασίας
<code>s = execve(όνομα, argv, environp)</code>	Αντικαθιστά την εικόνα πυρήνα μιας διεργασίας
<code>exit(κατάσταση)</code>	Τερματίζει την εκτέλεση της διεργασίας και επιστρέφει τον κωδικό κατάστασης

- Κλήσεις διαχείρισης διεργασιών
  - `fork()`: δημιουργεί αντίγραφο της τρέχουσας διεργασίας
    - Επιστρέφει τον αριθμό του παιδιού στον πατέρα και 0 στο παιδί
  - `waitpid()`: αναμονή μέχρι να τερματίσει ένα παιδί
    - Επιστρέφει αριθμό διεργασίας και κατάσταση εξόδου
  - `execve()`: αντικαθιστά τον κώδικα μιας διεργασίας
  - `exit ()`: τερματισμός τρέχουσας διεργασίας

# Διαχείριση διεργασιών (2 από 3)

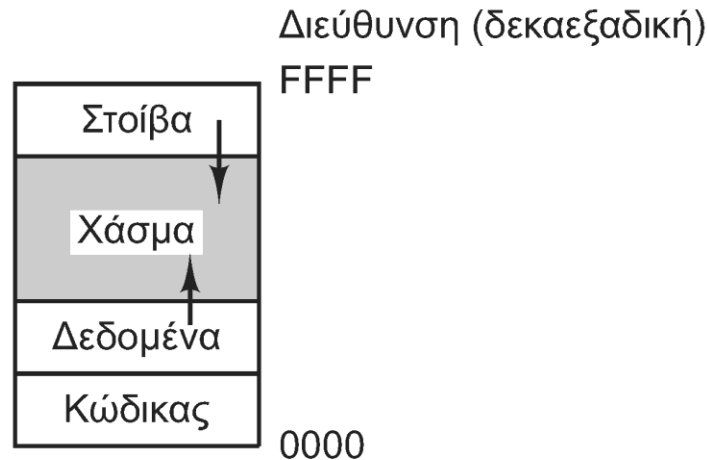
```
#define TRUE 1

while (TRUE) {
    type_prompt();
    read_command(command, parameters);
    if (fork() != 0) {
        /* κώδικας γονικής διεργασίας */
        waitpid(-1, &status, 0);
    } else {
        /* κώδικας θυγατρικής διεργασίας */
        execve(command, parameters, 0);
    }
}
```

*/\* επανάληψη συνεχώς \*/*  
*/\* εκτύπωση του προτροπτικού μηνύματος \*/*  
*/\* στην οθόνη \*/*  
*/\* ανάγνωση της εισόδου από το τερματικό \*/*  
*/\* δημιουργία θυγατρικής διεργασίας \*/*  
*/\* αναμονή ολοκλήρωσης θυγατρικής \*/*  
*/\* εκτέλεση της εντολής \*/*

- Παράδειγμα: ένας απλός φλοιός (shell)
  - Ανάγνωση εντολής, δημιουργία παιδιού για εκτέλεση
  - Ο πατέρας περιμένει το παιδί επαναλαμβάνει

# Διαχείριση διεργασιών (3 από 3)



- Χάρτης μνήμης διεργασίας στο UNIX
  - Κώδικας, δεδομένα και στοίβα
- Παράμετροι εκτέλεσης διεργασιών (`argc`, `argv`, `envp`)
  - `argc`, `argv`: πλήθος παραμέτρων και δείκτες σε αυτές
  - `envp`: δείκτες σε μεταβλητές περιβάλλοντος

# Διαχείριση αρχείων (1 από 4)

Κλήση	Περιγραφή δράσης
<code>fd = open(αρχείο, τρόπος, ...)</code>	Ανοίγει ένα αρχείο για ανάγνωση, εγγραφή, ή και τα δύο
<code>s = close(fd)</code>	Κλείνει ένα ανοιχτό αρχείο
<code>n = read(fd, buffer, nbytes)</code>	Διαβάζει δεδομένα από ένα αρχείο και τα τοποθετεί σε προσωρινή μνήμη
<code>n = write(fd, buffer, nbytes)</code>	Διαβάζει δεδομένα από μια προσωρινή μνήμη και τα αποθηκεύει σε ένα αρχείο
<code>position = lseek(fd, απόσταση, whence)</code>	Μετακινεί το δείκτη αρχείου
<code>s = stat(όνομα, &amp;buf)</code>	Διαβάζει τις πληροφορίες κατάστασης ενός αρχείου

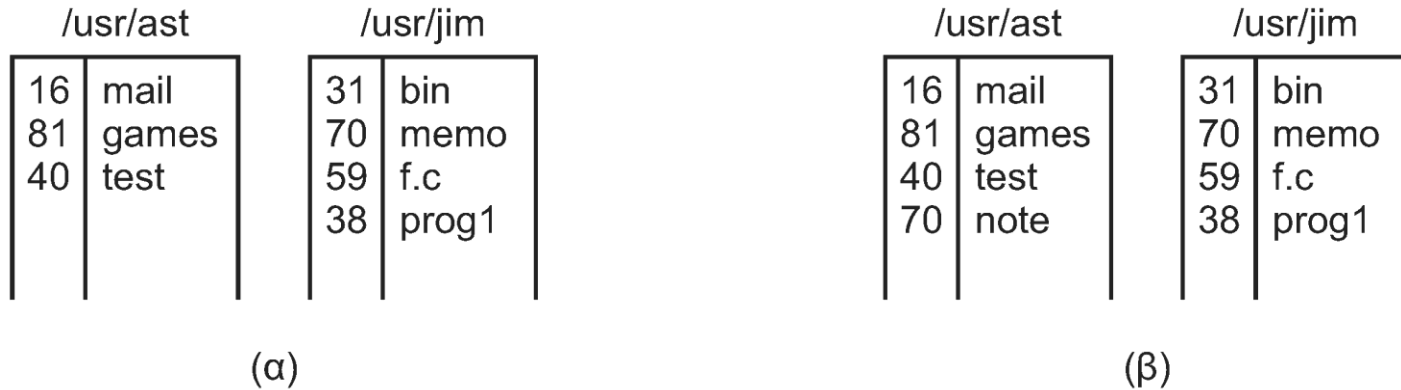
- Κλήσεις διαχείρισης αρχείων
  - `open()`: άνοιγμα ή δημιουργία αρχείου
  - `close()`: κλείσιμο αρχείου
  - `read()/write()`: ανάγνωση/εγγραφή αρχείου
  - `lseek()`: μετακίνηση σημείου ανάγνωσης/εγγραφής
  - `stat()`: ανάγνωση μεταδεδομένων αρχείου

# Διαχείριση αρχείων (2 από 4)

Κλήση	Περιγραφή δράσης
s = mkdir(όνομα, κατάσταση)	Δημιουργεί ένα νέο κατάλογο
s = rmdir(όνομα)	Διαγράφει έναν κενό κατάλογο
s = link(όνομα1, όνομα2)	Δημιουργεί μια νέα καταχώριση όνομα2 η οποία δείχνει στο όνομα1
s = unlink(όνομα)	Διαγράφει μια καταχώριση καταλόγου
s = mount(ειδική, όνομα, σημαία)	Αναρτά ένα σύστημα αρχείων
s = umount(ειδική)	Αποαναρτά ένα σύστημα αρχείων

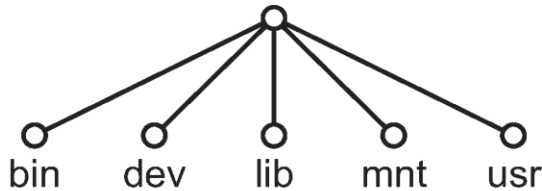
- Διαχείριση καταλόγων
  - mkdir(): δημιουργία νέου καταλόγου
  - rmdir(): διαγραφή καταλόγου (πρέπει να είναι κενός)
  - link(): δημιουργία συνδέσμου προς αρχείο/κατάλογο
  - unlink(): διαγραφή συνδέσμου ή αρχείου
  - (u)mount(): ανάρτηση/απομάκρυνση συστήματος αρχείων

# Διαχείριση αρχείων (3 από 4)

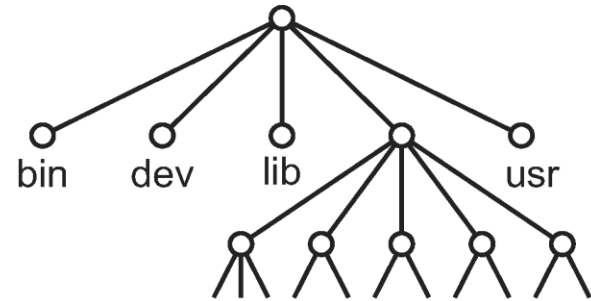


- Λειτουργία των συνδέσεων
  - Παράδειγμα: `link("/usr/jim/memo", "/usr/ast/note");`
  - Στον `/usr/ast` εμφανίζεται το αρχείο `/usr/jim/memo`
  - Στο UNIX κάθε αρχείο αντιπροσωπεύεται από έναν κόμβο `i`
  - Η `link()` εισάγει δείκτη προς υπάρχοντα κόμβο `i`
  - Το αρχείο διαγράφεται όταν γίνει `unlink()` από παντού

# Διαχείριση αρχείων (4 από 4)



(α)



(β)

- Λειτουργία της ανάρτησης
  - `mount("/dev/hda", "/mnt", 0);`
  - Ανάρτηση του σκληρού δίσκου `/dev/hda` κάτω από το `/mnt`
  - Το σύστημα αρχείων του δίσκου είναι ορατό
  - Το σύστημα ξεκινάει με ένα ριζικό σύστημα αρχείων
  - Στη συνέχεια αναρτώνται πρόσθετα συστήματα αρχείων



# Διάφορες κλήσεις

Κλήση	Περιγραφή δράσης
s = chdir(όνομακαταλόγου)	Αλλάζει τον κατάλογο εργασίας
s = chmod(όνομα, κατάσταση)	Αλλάζει τα bit προστασίας ενός αρχείου
s = kill(pid, σήμα)	Στέλνει σήμα σε μια διεργασία
seconds = time(&seconds)	Υπολογίζει το χρόνο που έχει περάσει από την 1/1/1970

- chdir(): αλλαγή καταλόγου εργασίας
- chmod(): αλλαγή προνομίων πρόσβασης
  - Read/write/execute για user/group/others
- kill(): αποστολή σήματος σε διεργασία
  - Τα σήματα που δεν συλλαμβάνονται σκοτώνουν τον παραλήπτη
- time(): επιστρέφει την τρέχουσα ώρα
  - Χρονικό διάστημα από μια σταθερή χρονική στιγμή

# Windows και UNIX

- Το μοντέλο προγραμματισμού Windows διαφέρει
  - Επικεντρώνεται στα γεγονότα από το περιβάλλον
- Η βασική διεπαφή είναι το Win32 API
  - Η διεπαφή χρησιμοποιείται σε πολλές εκδόσεις Windows
  - Η ανάγκη συμβατότητας κάνει τη διεπαφή ασυνεπή
  - Τα Windows έχουν πολλά API και χιλιάδες κλήσεις
    - Πολλές κλήσεις δεν αντιστοιχούν σε κλήσεις συστήματος
- Ορισμένες διαφορές των Windows από το UNIX
  - Η δημιουργία νέας διεργασίας θέλει ένα μόνο βήμα
  - Παρέχεται μία κλήση για αναμονή πολλών συμβάντων

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Δομή ΛΣ

**Μάθημα:** Λειτουργικά Συστήματα, **Ενότητα # 1:** Εισαγωγή

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Δομή ΛΣ (1 από 7)

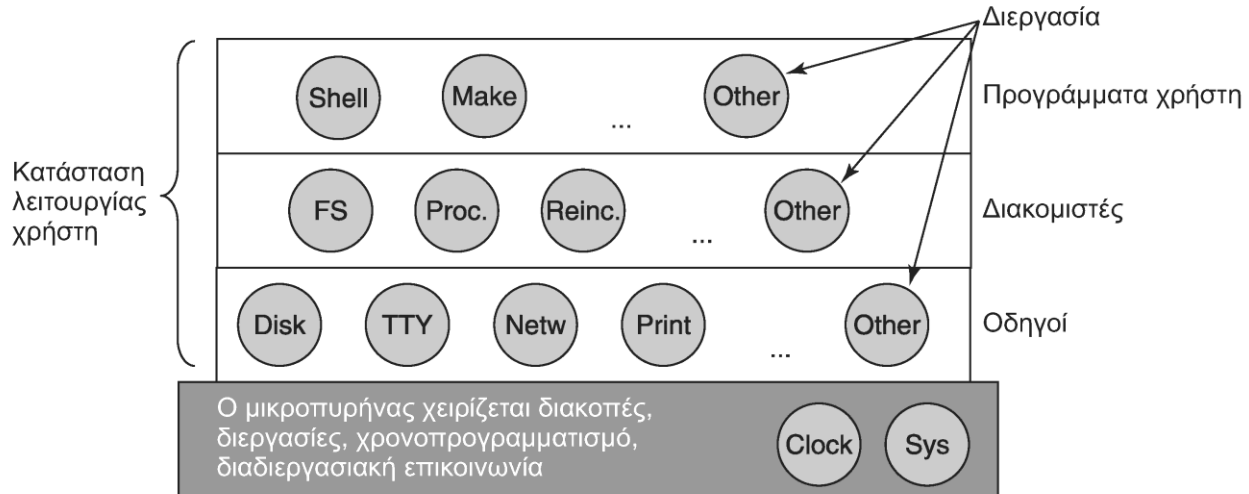
- Μονολιθικά συστήματα
  - Όλο το ΛΣ είναι ένα τεράστιο πρόγραμμα
  - Το σύνολο του ΛΣ εκτελείται σε κατάσταση πυρήνα
  - Κάθε διαδικασία μπορεί να καλέσει κάθε άλλη
    - Τα πάντα είναι ορατά σε όλους (δομές, διαδικασίες)
    - Δυσκολία στη συντήρηση και στη διόρθωση σφαλμάτων
  - Στοιχειώδης οργάνωση του κώδικα
    - Η είσοδος στο ΛΣ γίνεται με μια παγίδα
    - Κύριο πρόγραμμα που καλεί την αντίστοιχη διαδικασία
    - Διαδικασίες που εξυπηρετούν τις κλήσεις συστήματος

# Δομή ΛΣ (2 από 7)

Επίπεδο	Λειτουργία
5	Χειριστής
4	Προγράμματα χρηστών
3	Διαχείριση εισόδου/εξόδου
2	Επικοινωνία χειριστή-διεργασίας
1	Διαχείριση μνήμης και τυμπάνου
0	Εκχώρηση επεξεργαστή και πολυπρογραμματισμός

- Πολυεπίπεδα συστήματα
  - Ιδέα του E.W. Dijkstra που εφαρμόστηκε στο THE
  - Κάθε επίπεδο παρέχει υπηρεσίες στα παραπάνω επίπεδα
  - Γενικεύθηκε στο σύστημα MULTICS
    - Οργάνωση σε ομόκεντρους δακτυλίους
    - Συνδυασμός δόμησης σε επίπεδα και προστασίας

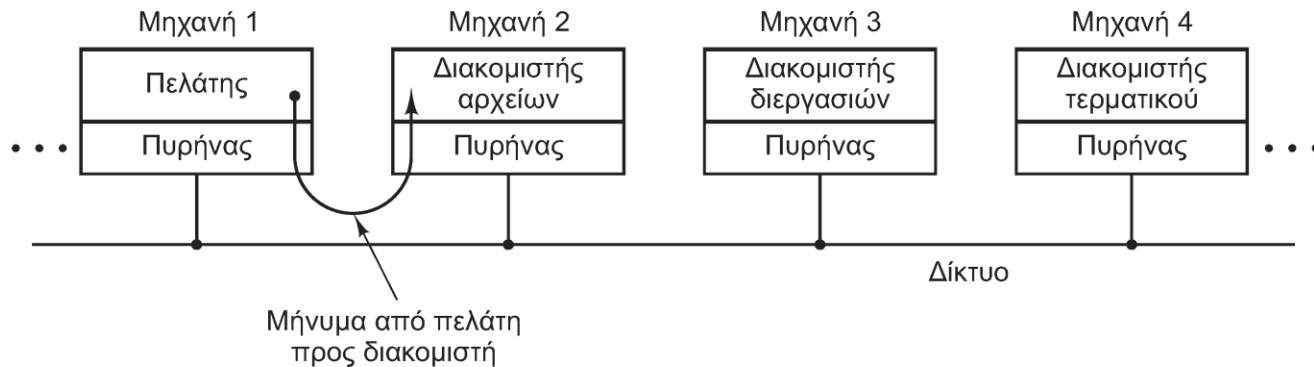
# Δομή ΛΣ (3 από 7)



- Μικροπυρήνες

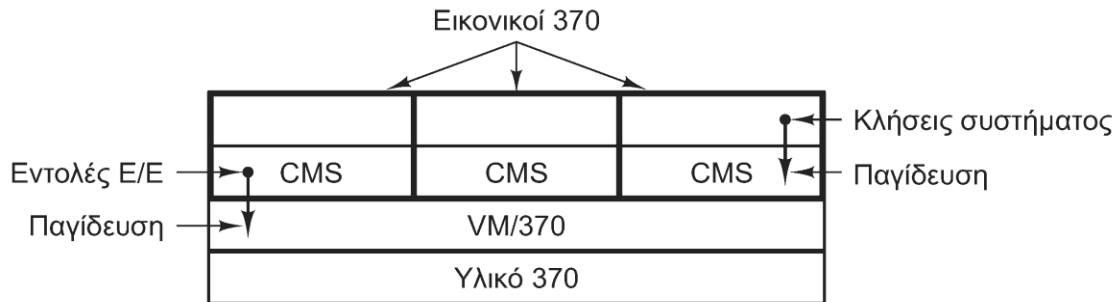
- Ελάχιστο σύνολο λειτουργιών σε κατάσταση επόπτη
  - Διακοπές, διεργασίες, χρονοπρογραμματισμός, IPC
- Οι υπόλοιπες εκτελούνται σε κατάσταση χρήστη
- Επικοινωνία μέσω μηνυμάτων ή IPC
- Μείωση πιθανότητας κατάρρευσης λόγω σφαλμάτων

# Δομή ΛΣ (4 από 7)



- Μοντέλο πελάτη-εξυπηρετητή
  - Εξυπηρετητές: παρέχουν υπηρεσίες
  - Πελάτες: ζητούν υπηρεσίες από τους εξυπηρετητές
  - Επικοινωνία μέσω μεταβίβασης μηνυμάτων
    - Απλουστεύει την κατανομή υπηρεσιών σε δίκτυο

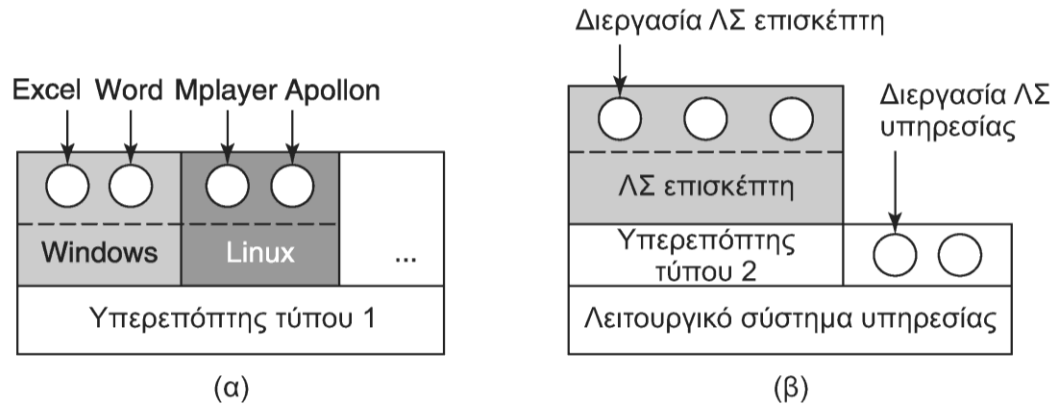
# Δομή ΛΣ (5 από 7)



- Εικονικές μηχανές
  - Ξεκίνησε με το VM/370 και επιβιώνει στο z/VM
  - Βασικός πυρήνας που εκτελείται πάνω στο υλικό
  - Παρέχει την εικόνα πολλών πανομοιότυπων μηχανών
    - Παγίδευση και εκτέλεση όλων των προνομιούχων εντολών
  - Οι μηχανές είναι ακριβή αντίγραφα του υλικού
    - Αρχικά, συστήματα δέσμης ή διαλογικά συστήματα ενός χρήστη
    - Σήμερα, πλήρη λειτουργικά συστήματα όπως Linux ή Windows



# Δομή ΛΣ (6 από 7)



- Η επιστροφή: εικονικοποίηση χωρίς ειδικό υλικό
  - VMware: δυαδική μετάφραση κώδικα του λειτουργικού
  - XEN: τροποποίηση του κώδικα του λειτουργικού
- Δύο τύποι υπερεπόπτη
  - Τύπου 1: εκτελείται απευθείας πάνω στο υλικό
  - Τύπου 2: εκτελείται μέσα σε ένα λειτουργικό σύστημα

# Δομή ΛΣ (7 από 7)

- Εικονική μηχανή Java
  - Εικονικοποίηση σε επίπεδο λογισμικού
  - Ο κώδικας της Java εκτελείται σε μια εικονική μηχανή
- Εξωπυρήνες
  - Διαμέριση των πόρων ανάμεσα σε εικονικές μηχανές
    - Κάθε χρήστης παίρνει μέρος του δίσκου και της μνήμης
  - Κάθε εικονική μηχανή γνωρίζει ότι έχει μέρος των πόρων
    - Στην εικονικοποίηση νομίζει ότι έχει ολόκληρη τη μηχανή
  - Πολύ απλούστερη υλοποίηση του εξωπυρήνα
    - Δεν χρειάζεται μετάφραση εικονικών σε φυσικούς πόρους

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Η γλώσσα C

**Μάθημα:** Λειτουργικά Συστήματα, **Ενότητα # 1:** Εισαγωγή

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

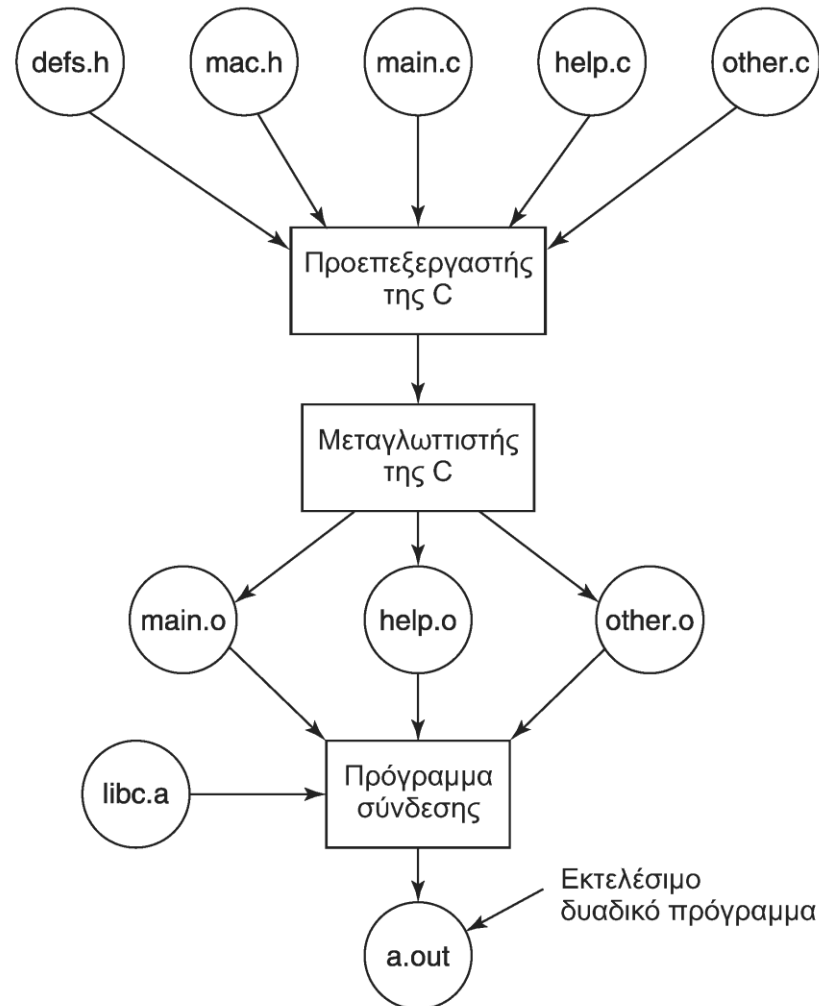
# Η γλώσσα C (1 από 3)

- Από την Java στην C σε λίγα λεπτά
  - Η Java μοιάζει αρκετά στη σύνταξη με τη C
  - Απλοί τύποι δεδομένων και δομές, πίνακες και ενώσεις
  - Συνηθισμένες εντολές ελέγχου (if, switch, for, while)
  - Δυνατότητα χρήσης δεικτών στη μνήμη
  - Οι δείκτες θεωρητικά έχουν τύπους (π.χ. σε ακέραιο)
  - Η C δεν κάνει αυτόματη διαχείριση μνήμης
    - Απαιτείται ρητή δέσμευση και αποδέσμευση μνήμης
    - Η δυναμική μνήμη χρησιμοποιείται μέσω δεικτών
  - Η C είναι πολύ ισχυρή, αλλά και πολύ επικίνδυνη

# Η γλώσσα C (2 από 3)

- Αρχεία κεφαλίδες
  - Περιέχουν δηλώσεις και ορισμούς τύπων / μακροεντολών
  - Οι αντικαταστάσεις γίνονται από τον προεπεξεργαστή C
    - Αντικατάσταση κειμένου, μεταγλώττιση υπό συνθήκη
- Μεταγλώττιση και σύνδεση
  - Ο μεταγλωττιστής καλεί πρώτα τον προεπεξεργαστή
  - Η έξοδος του μεταγλωττιστή είναι αντικειμενικό αρχείο
  - Τα αρχεία συνδέονται (στατικά ή δυναμικά) σε εκτελέσιμα
  - Χρήση `make` για μερική μεταγλώττιση συστήματος

# Η γλώσσα C (3 από 3)



**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**

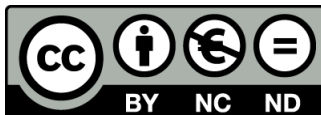


**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Τέλος Ενότητας #1

**Μάθημα:** Λειτουργικά Συστήματα, **Ενότητα # 1:** Εισαγωγή

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

