

# ΕΙΣΑΓΩΓΗ

## ΣΥΣΤΑΤΙΚΑ ΣΥΣΤΗΜΑΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

### Υλισμικό (Hardware)

- Επεξεργαστής, ΚΜΕ (Processor, CPU): εκτέλεση εντολών
- Κύρια Μνήμη (Main Memory): προσωρινή αποθήκευση πληροφορίας
- Βοηθητική Μνήμη (Auxiliary Memory):
  - Μονάδες δίσκων: Μόνιμη αποθήκευση πληροφορίας
- Ελεγκτές ε/ε (I/O controllers): μεταφορά από και προς συσκευές ε/ε

### ΛΟΓΙΣΜΙΚΟ (Software)

- Λειτουργικό Σύστημα, ΛΣ (Operating System, OS)
- Προγράμματα εφαρμογών (application programs)
  - όλα τα υπόλοιπα προγράμματα

### ΠΡΟΓΡΑΜΜΑΤΑ

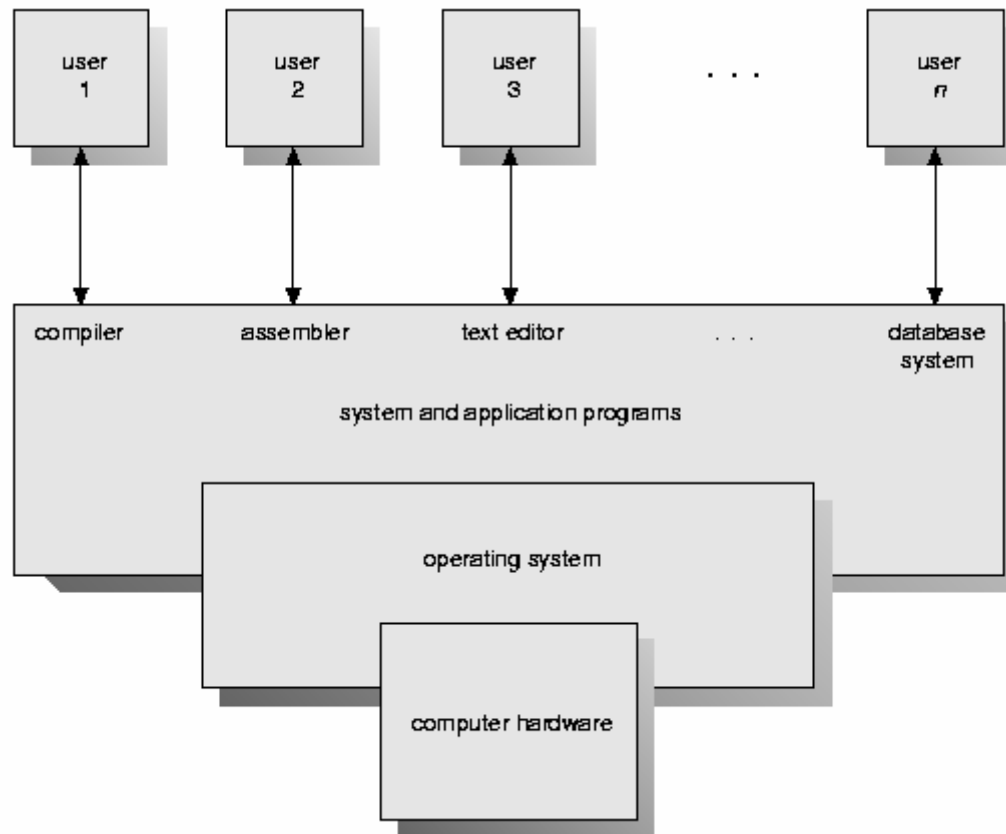
- Κώδικας (εντολές)
- Δεδομένα

### ΧΡΗΣΤΕΣ

# ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ

- Ένα (παράλληλο) πρόγραμμα που ελέγχει τη λειτουργία της μηχανής
- Τυπικά ένα ΛΣ σε κάθε υπολογιστή
- Η εκτέλεσή του αρχίζει όταν ανοίγει ο διακόπτης ρεύματος του υπολογιστή
- Μόνιμα φορτωμένο στην μνήμη - "τρέχει" συνεχώς

Παρέχει μια διεπαφή (interface) ανάμεσα στο υλισμικό και στα προγράμματα εφαρμογών



# ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

## ΠΡΩΤΑ ΣΥΣΤΗΜΑΤΑ (bare machines, open shop ή paper tape systems)

### Χαρακτηριστικά

- Τεράστιες μηχανές με πίνακα ελέγχου
- Χρήστης/προγραμματιστής = χειριστής
- ε/ε: χαρτοταινίες
- Λογισμικό: Συμβολομεταφραστές, φορτωτές
- Χειρωνακτικές διαδικασίες από το χρήστη (BOOT, LOAD, RUN)
- Λειτουργία εξαρτώμενη από το χρήστη = από-γραμμής (off-line)
- μεγάλος χρόνος ανακύκλωσης, μικρή ρυθμαπόδοση

### χρόνος ανακύκλωσης (turnaround time)

χρόνος παραλαβής αποτελεσμάτων – χρόνος υποβολής εργασίας

### ρυθμαπόδοση (throughput)

πλήθος εργασιών που εκτελούνται ανά μονάδα χρόνου

# ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

## ΣΥΣΤΗΜΑΤΑ ΧΕΙΡΙΣΤΗ (operator driven shop)

- Χειριστής  $\neq$  χρήστης
- ❖ Αναγνώστες και διατρητές δελτίων (καρτών), εκτυπωτές γραμμών
- ❖ Μαγνητικές ταινίες για την αποθήκευση των ενδιάμεσων αρχείων
- ❖ Βιβλιοθήκες με κοινόχρηστες ρουτίνες
- ❖ Μεταγλωττιστές, συνδέτες
- ❖ Ακολουθιακή εκτέλεση παρόμοιων εργασιών πολλών χρηστών
- ❖ Λειτουργία εξαρτώμενη από το χειριστή (προτεραιότητες, διακοπή εργασίας)
- Καλύτερος χρόνος ανακύκλωσης και ρυθμαπόδοση

# ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

## ΑΚΟΛΟΥΘΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΔΕΣΜΙΔΩΝ (BATCH SYSTEMS)

- ❖ Πρόγραμμα παρακολούθησης ή επόπτης
- ❖ (monitor/supervisor/executive/director/master control program)→ΛΣ
- ❖ μόνιμα φορτωμένο στη μνήμη (resident monitor)
- ❖ Γλώσσα ελέγχου εργασιών
  - Ο παρακολουθητής περιλαμβάνει το διερμηνευτή καρτών ελέγχου
  - Ο χειριστής δημιουργεί δεσμίδες εργασιών διαφόρων χρηστών

π.χ. δεσμίδα =

\$JOB | ... | \$FTN | ... | \$RUN | ... | \$DATA | ... | \$END

\$JOB | ... | \$ASS | ... | \$RUN | ... | \$DATA | ... | \$END

\$JOB | ... | \$COB | ... | \$RUN | ... | \$DATA | ... | \$END|\$EOB

- \$ Κάρτες ελέγχου
- Αυτόματη μετάβαση από μία εργασία σε άλλη
- Λειτουργία χωρίς την επέμβαση του χειριστή =επί-γραμμής (on-line)
- Ακόμη καλύτερος χρόνος ανακύκλωσης και ρυθμαπόδοση

## ΠΡΟΒΛΗΜΑ

- Η ΚΜΕ είναι γρήγορη αλλά και ακριβή
- οι αναγνώστες καρτών και οι εκτυπωτές είναι πολύ αργοί
- η ΚΜΕ έπρεπε να περιμένει για κάθε εργασία (παρέμενε αδρανής)

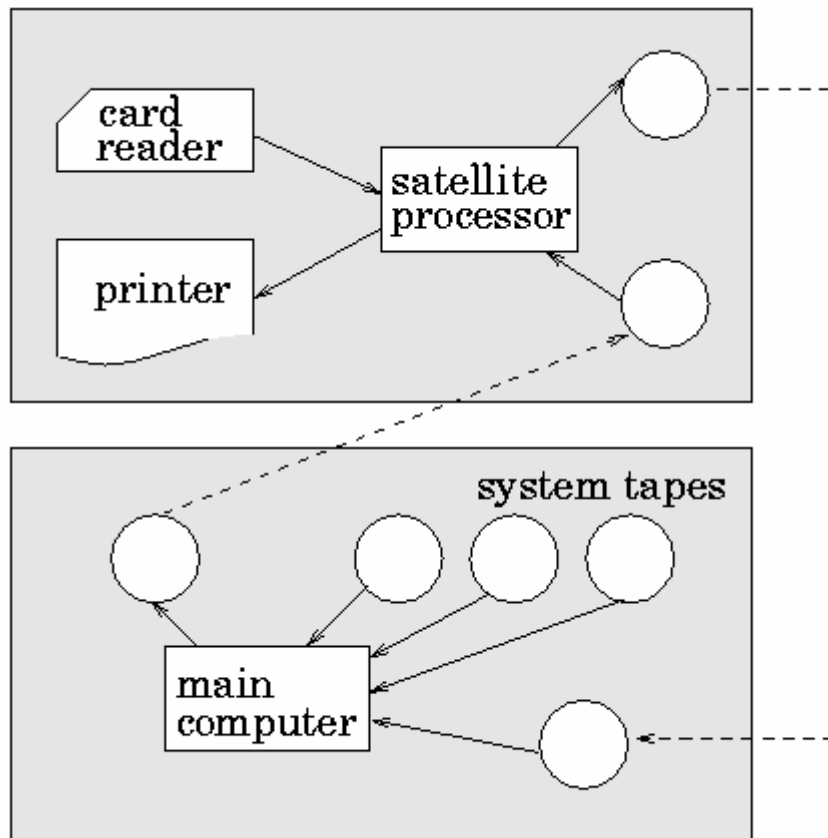
IN1	RUN 1	OUT1	IN2	RUN2	OUT2	IN3	RUN3	OUT3
-----	-------	------	-----	------	------	-----	------	------

# ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

## ΕΝΤΑΜΙΕΥΣΗ (BUFFERING)

**ΙΔΕΑ:** Επιτάχυνση ε/ε με ενδιάμεση αποθήκευση σε πιο γρήγορα μέσα

- ❖ **Ενταμίευση (buffering)** = ενδιάμεση αποθήκευση
  - επεξεργαστές δορυφόροι μεταφέρουν προγράμματα
    - από αναγνώστες καρτών προς μαγνητικές ταινίες και
    - από μαγνητικές ταινίες προς εκτυπωτή
  - ο χειριστής τις μεταφέρει στον κεντρικό επεξεργαστή



- Από-γραμμής (off-line) λειτουργία
- Προγράμματα: δε χρειάζεται καμιά μετατροπή για την αλλαγή από on-line σε off-line λειτουργία

## **ΠΡΟΒΛΗΜΑΤΑ**

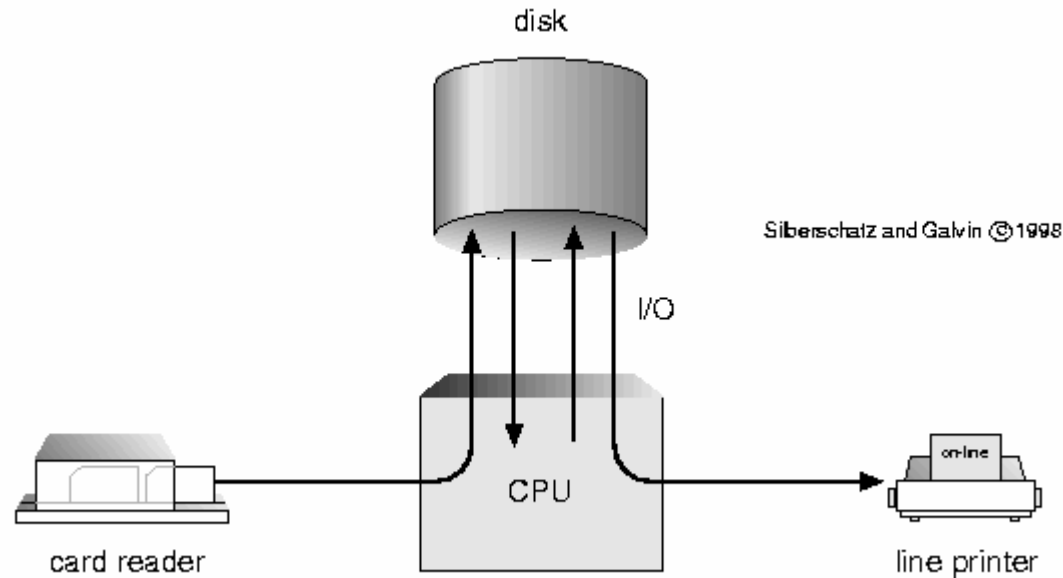
- Και πάλι η ΚΜΕ περιορίζεται - τώρα από την ταχύτητα των ταινιών
- Χειρωνακτική μεταφορά

# ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

## SPOOLING (Simultaneous Peripheral Operations On-Line) [ΤΑΥΤΟΧΡΟΝΕΣ ΕΠΙ-ΓΡΑΜΜΙΚΕΣ ΠΕΡΙΦΕΡΕΙΑΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ]

Επικάλυψη εκτέλεσης και ε/ε διαφόρων εργασιών

- ❖ Μαγνητικοί δίσκοι!
- ❖ Ο δίσκος αποτελεί ένα τεράστιο ενταμιευτή (για είσοδο και έξοδο)
- ❖ Αυτόματη ενταμίευση από την ίδια την ΚΜΕ, "ταυτόχρονα" με την εκτέλεση άλλων εργασιών= spooling



- Το σύστημα "ταυτόχρονα"
  - Διαβάζει από τον αναγνώστη δελτίων την επόμενη εργασία και την αποθηκεύει σε μια περιοχή του δίσκου, που ονομάζονται ουρά εργασιών (job queue)
  - Τυπώνει τα αποτελέσματα προηγούμενων εργασιών από το δίσκο στον εκτυπωτή
  - Εκτελεί μια άλλη εργασία που έχει επιλεγεί από την ουρά εργασιών στο δίσκο (δίσκος: δυνατότητα επιλογής)

▪ "ταυτόχρονα" = Ο παρακολουθητής (η ΚΜΕ) ξεκινάει τις μεταφορές και ειδοποιείται όταν μια μεταφορά τελειώσει

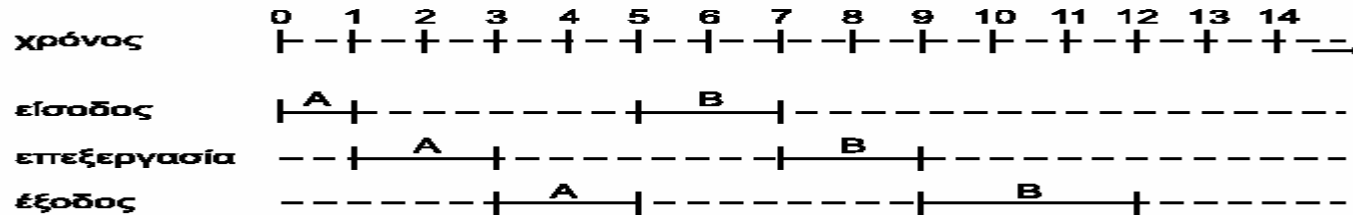
❖ **ΠΟΛΥΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ (MULTIPROGRAMMING)**

# ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

## ΠΟΛΥΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

### Παράδειγμα

#### 1) ΑΚΟΛΟΥΘΙΑΚΗ ΕΚΤΕΛΕΣΗ



$$\text{ρυθμαπόδοση} = 2/12 = 0.167$$

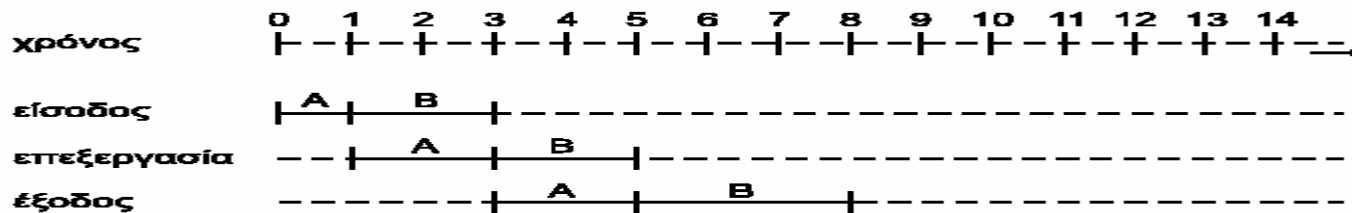
$$\text{Χρήση Εισόδου} = 3/12 = 25\%$$

$$\text{Χρήση ΚΜΕ} = 4/12 = 33.3\%$$

$$\text{Χρήση Εξόδου} = 5/12 = 41.7\%$$

#### 2) ΠΟΛΥΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

- Επικάλυψη εκτέλεσης και ε/ε διαφόρων εργασιών



$$\text{ρυθμαπόδοση} = 2/8 = 0.25$$

Χρήση Εισόδου =  $3/8 = 37.5\%$

Χρήση ΚΜΕ =  $4/8 = 50\%$

Χρήση Εξόδου =  $5/8 = 62.5\%$

# ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

## ΠΟΛΥΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

- Επικάλυψη εκτέλεσης και ε/ε διαφόρων εργασιών
- Πολλές εργασίες βρίσκονται ταυτόχρονα στη μνήμη
- Πόσες /Ποιες εργασίες? + Διαχείριση μνήμης
- Ποια εργασία θα εκτελεστεί? + Χρονοπρογραμματισμός (scheduling)

## ΠΡΟΒΛΗΜΑ

- Μια εργασία μπορεί να χρειαστεί να περιμένει για κάτι, π.χ.
  - Αλληλεπίδραση με χρήστη
  - Ταχύτητα δίσκων  $\ll$  ταχύτητα ΚΜΕ
- κατά τη διάρκεια της αναμονής η ΚΜΕ παραμένει αδρανής (idle)

ΙΔΕΑ: Εάν μια εργασία χρειαστεί να περιμένει για οτιδήποτε τότε η ΚΜΕ παραχωρείται σε μια άλλη εργασία

- ❖ Τερματικά (οθόνη - πληκτρολόγιο)
- ❖ Διακοπές (interrupts) : σημαίνουν το τέλος ε/ε
- ❖ Ρουτίνες ε/ε υποστηριζόμενες από το σύστημα (για τη λειτουργία των οδηγών συσκευών (device drivers))

## ΠΡΟΒΛΗΜΑ

Μια εργασία μπορεί να μονοπωλήσει την ΚΜΕ

# ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

## (Κατα)μερισμός χρόνου (Time-sharing)

**Παραχώρηση συγκεκριμένου χρονικού διαστήματος (time-slice) σε κάθε εργασία**

- μετά την παρέλευση του διαστήματος η εργασία διακόπτεται για να συνεχιστεί αργότερα
- (ψευδ)αίσθηση ότι ο κάθε χρήστης έχει δικό του υπολογιστή
- Λειτουργία επιγραμμής (on-line)
- ❖ Γλώσσα εντολών ΛΣ (command language)
- ❖ Διερμηνευτής εντολών (φλοιός)
- ❖ On-line σύστημα αρχείων
- ❖ .....

# ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

## UNIX

- ❖ Παραθυρικά περιβάλλοντα (X-terminals,...)

## Προσωπικοί υπολογιστές (personal computers)

- Ένας χρήστης (= χειριστής =προγραμματιστής)  
(όπως στα πρώτα συστήματα, πλήρης κύκλος)
  - Πολλές εφαρμογές έτοιμες να εκτελεστούν / εκτελούνται την "ίδια" στιγμή
- ΑΛΛΑ ΚΑΙ ΠΑΛΙ: Πολύ χαμηλή χρήση της ΚΜΕ

# ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

## Κατανεμημένα συστήματα (Distributed systems)

- Κατανομή του φόρτου εργασίας σε PC's/WSs διασυνδεδεμένων με LANs - Internet
- Κατανομή πόρων
- Επιτάχυνση επεξεργασίας
- Αξιοπιστία
- Επεκτασιμότητα
- Χαλαρά συνδεδεμένα συστήματα (loosely coupled systems)

## Παράλληλα συστήματα (Parallel systems)

- Συστήματα πολλών επεξεργαστών /ΚΜΕς (multiprocessor systems)
- Επικοινωνία μέσω κοινής/(κατά)μεριζόμενης μνήμης (shared memory)
- Σφικτά συνδεδεμένα συστήματα (tightly coupled systems)

## Συστήματα πραγματικού χρόνου (Real-time systems)

- Αυστηροί χρονικοί περιορισμοί προθεσμίας (deadlines) απόκρισης
- Μονάδες ελέγχου άλλων συστημάτων

# ΤΙ ΕΙΝΑΙ ΕΝΑ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ

## Διαχειριστής πόρων (resource manager)

- Σύνολο αλγορίθμων που εκχωρούν πόρους στις διεργασίες
- Στόχος: αποτελεσματικότητα

## αρχή πόρων (resource principle)

## Δημιουργός εικονικών/διευρυμένων/λογικών πόρων (virtual/extended/logical resources)

- Σύνολο αλγορίθμων που παρέχουν με λογική αφαίρεση (logical abstraction) ένα πιο ευχάριστο περιβάλλον από το γυμνό υλισμικό

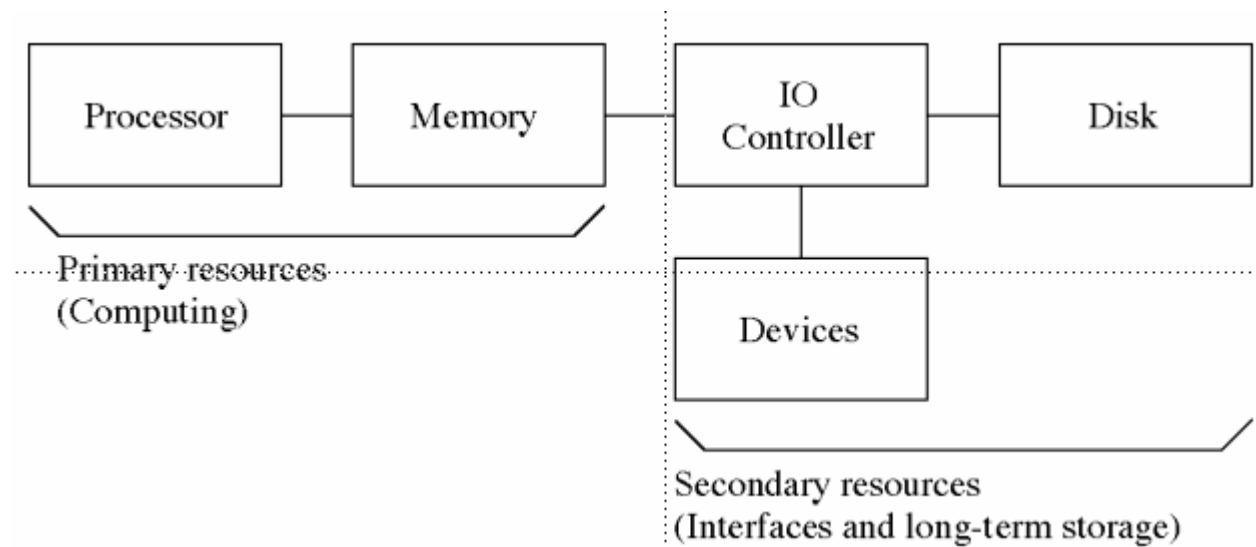
## αρχή εξωραϊσμού (beautification principle)

- κρύβει τις λεπτομέρειες από τον χρήστη
- δημιουργεί λογικές έννοιες/πόρους (αρχείο, διεργασία, ...)
- ομογενοποιεί την αντιμετώπιση διαφορετικών αντικειμένων του υλισμικού (π.χ. μια μοναδική λειτουργία read)

# ΦΥΣΙΚΟΙ ΠΟΡΟΙ

## Παρέχονται στο ΛΣ

- Επεξεργαστής - ΚΜΕ (Processor - CPU): εκτέλεση εντολών
- Μνήμη (Memory): Προσωρινή αποθήκευση πληροφορίας
- Μονάδες Ε/Ε (I/O devices):
  - Μονάδες δίσκων: μόνιμη αποθήκευση πληροφορίας
  - Άλλες μονάδες
- Ελεγκτές Ε/Ε (I/O controllers): μεταφορά από και προς μονάδες ε/ε



Crowley OS

- **Σειριακά επαναχρησιμοποιούμενοι πόροι (serially-reusable resources)**  
ΚΜΕ, Εκτυπωτής, ...
- **Πόροι πολλαπλής χρήσης (multiple-use resources)**  
Μνήμη, δίσκος, ...

# ΕΙΚΟΝΙΚΟΙ-ΛΟΓΙΚΟΙ ΠΟΡΟΙ-ΜΗΧΑΝΕΣ

Δημιουργούνται από το ΛΣ

Φυσική διεπαφή → Εικονική διεπαφή

## Μετατροπή (transformation)

- ενός φυσικού πόρου σε ένα *στιγμιότυπο* εικονικού/λογικού πόρου
- αφαίρεση, απόκρυψη λεπτομέρειας
- π.χ. τομέας-ομάδα στο δίσκο → αρχείο

## Πολύπλεξη (multiplexing)

- δημιουργία *πολλαπλών στιγμιότυπων* εικονικών πόρων ενός φυσικού πόρου
- **Πολύπλεξη διαίρεσης χρόνου (Time Division Multiplexing, TDM)**
  - (Κατά)μερισμός χρόνου στις εργασίες
  - Σειριακά επαναχρησιμοποιούμενοι πόροι
    - π.χ. εκτυπωτής → SPOOLing αρχεία
- **Πολύπλεξη διαίρεσης χώρου (Space Division Multiplexing, SDM)**
  - (Κατά)μερισμός χώρου στις εργασίες
  - Πόροι πολλαπλής χρήσης
  - π.χ. δίσκος → αρχεία

## Χρονοπρογραμματισμός (Scheduling)

- Ποια διεργασία θα χρησιμοποιήσει τους πόρους

# ΕΙΚΟΝΙΚΟΙ-ΛΟΓΙΚΟΙ ΠΟΡΟΙ-ΜΗΧΑΝΕΣ

Το ΛΣ δημιουργεί για κάθε εργασία/χρήστη

- μια πλήρη εικονική μηχανή
- την αυταπάτη ότι έχει την δική του μηχανή
- ΚΜΕ → διεργασίες (processes) - TDM
- Μνήμη → Χώρος διευθύνσεων (address space) - TDM + SDM
- Δίσκος → αρχεία - Transformation + SDM
- Ε/Ε → Συσκευές/Ενταμιευτές - Transformation + TDM

# ΛΕΙΤΟΥΡΓΙΑ ΣΥΣΤΗΜΑΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

- ΚΜΕ και μονάδες ε/ε λειτουργούν ταυτόχρονα
- Κάθε ελεγκτής συσκευών είναι υπεύθυνος για ένα συγκεκριμένο τύπο μονάδων
- Κάθε ελεγκτής συσκευών έχει το δικό του τοπικό ενταμιευτή
- Η ΚΜΕ μεταφέρει πληροφορία από/προς την μνήμη προς/από τους τοπικούς ενταμιευτές
- Η ε/ε γίνεται από/προς τις συσκευές προς/από τους τοπικούς ενταμιευτές
- Ο ελεγκτής συσκευών πληροφορεί την ΚΜΕ όταν τελειώσει την μεταφορά του με μια διακοπή

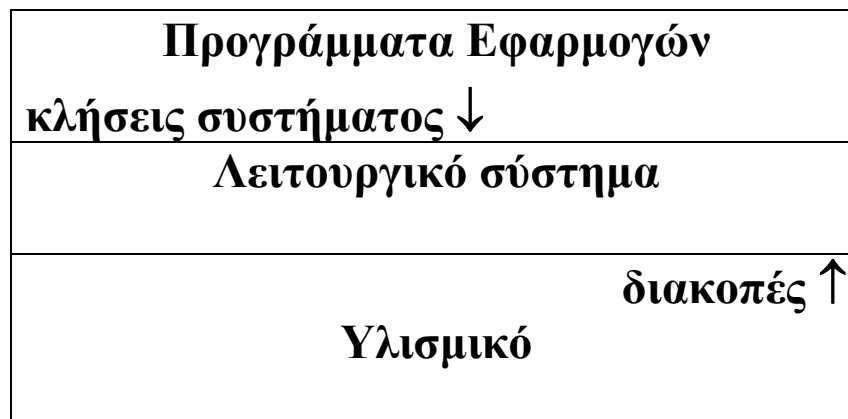
# ΛΕΙΤΟΥΡΓΙΑ ΣΥΣΤΗΜΑΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

## Δυϊκός τρόπος λειτουργίας

- Τρόπος χρήστη (user mode)
- Τρόπος συστήματος (system mode)
- mode bit στο υλισμικό (0: system, 1: user)
- Προνομιούχες (privileged) εντολές εκτελούνται μόνο στην κατάσταση του συστήματος

## Κλήσεις συστήματος (system calls)

- `syscall`, `trap`, `SVC`
- Επικοινωνία εφαρμογής/χρήστη με το ΛΣ
- Σφάλμα (fault) ή αίτηση εφαρμογής/χρήστη
- Εντολές συμβολικών γλωσσών
- Μερικές γλώσσες υψηλού επιπέδου παρέχουν κατευθείαν κλήσεις του συστήματος (C, Bliss, PL/360 κ.ά.)



## ΥΠΗΡΕΣΙΕΣ ΛΣ

- Εκτέλεση προγραμμάτων
- Λειτουργίες ε/ε
- Διαχείριση συστήματος αρχείων
- Εντοπισμός σφαλμάτων
- Καταχώριση πόρων
- Λογιστική χρέωση
- Προστασία
- Διεπαφή χρήστη
- Διερμηνεία εντολών (command interpretation)
- ...

# ΣΥΣΤΑΤΙΚΑ ΕΝΟΣ ΛΣ

## ▪ Πυρήνας (kernel)

- Διαχείριση διακοπών
- Δημιουργία, συγχρονισμός και επικοινωνία διεργασιών
- (Κατά)μερισμός χρόνου
- Μέτρηση χρήσης των πόρων

## ▪ Προγράμματα συστήματος

- Διαχειριστές εισόδου/εξόδου (I/O managers)
- Διαχειριστής μνήμης (memory manager)
- Διαχειριστής ΚΜΕ (CPU manager)
- Διαχειριστής συστήματος αρχείων (file system manager)
- Μηχανισμοί προστασίας (protection)
- Διεπαφή χρήστη (user interface)

# ΣΧΕΔΙΑΣΜΟΣ ΛΣ

## Σχεδιασμός σε επίπεδα (layered design)

Χρήστης		
Διεπαφή χρήστη		
	Προγράμματα συστήματος	
...		...
Πυρήνας		
Υλισμικό		

- Κάθε επίπεδο χρησιμοποιεί λειτουργίες μόνο των κατώτερων επιπέδων
- Γενικά δύο επίπεδα
  - Χαμηλότερο επίπεδο = *Μηχανισμοί (mechanisms)*
    - πώς θα γίνει κάτι
    - δεν αλλάζει - εξαρτώμενο ισχυρά από υλισμικό
    - χρησιμοποιήσιμο από διαφορετικά υψηλότερα επίπεδα
  - Υψηλότερο επίπεδο = *Πολιτικές (policies)*
    - Τι θα γίνει
    - Αλλάζει αργότερα
    - διαφορετικά υψηλότερα επίπεδα μπορούν να χρησιμοποιήσουν το ίδιο χαμηλότερο επίπεδο

# ΣΤΟΧΟΙ ΣΧΕΔΙΑΣΜΟΥ ΛΣ

## Στόχοι από την πλευρά του χρήστη

- Εύκολο στη εκμάθηση και τη χρήση
- Αξιόπιστο
- Ασφαλές
- Γρήγορο

## Στόχοι από την πλευρά του συστήματος

- Εύκολο στη σχεδίαση και συντήρηση
- Ευέλικτο
- Αξιόπιστο
- Χωρίς λάθη
- Αποτελεσματικό

# ΥΛΟΠΟΙΗΣΗ ΛΣ

- Παραδοσιακά σε συμβολική γλώσσα
- Σήμερα και σε γλώσσες υψηλού επιπέδου
  - Πιο γρήγορη υλοποίηση
  - Περισσότερο συμπαγές
  - Ευκολότερη κατανόηση και εκσφαλμάτωση
  - Φορητότητα