ΣΥΓΧΡΟΝΑ ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

3η έκδοση

ANDREW S. TANENBAUM

ΚΕΦΑΛΑΙΟ 1ο
Εισαγωγή

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης

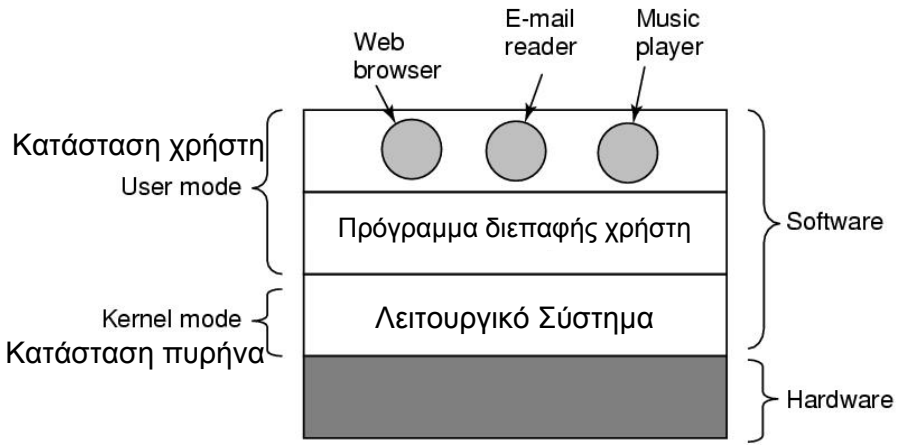
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Κεφάλαιο 1ο

Περιεχόμενα

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Εισαγωγή
Δομή υπολογιστικού συστήματος



Εικόνα 1-1. Που βρίσκεται το ΛΣ

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

- Πολλές φορές γίνεται σύγχυση μεταξύ του ίδιου του ΛΣ και του συστήματος αλληλεπίδρασης με αυτό (command shell, GUI)
- Το ΛΣ βρίσκεται σε **κατάσταση πυρήνα (kernel mode):** – δεν μπορεί να τροποποιηθεί από τον προγραμματιστή.
    - Οι εντολές ελέγχου της μηχανής και οι εντολές Ε/Ε, είναι διαθέσιμες μόνο σε kernel mode.
- Τα υπόλοιπο λογισμικό βρίσκεται σε **κατάσταση χρήστη (user mode):** – μπορεί να έχει πρόσβαση στην εκτέλεσή του ο προγραμματιστής.
    - Στην κατάσταση χρήστη είναι διαθέσιμο μόνο ένα υποσύνολο των εντολών μηχανής και όχι όλες.
- Ο προγραμματιστής μπορεί να αλλάξει / γράψει νέα προγράμματα στο user space αλλά **όχι στο kernel space** (δεν μπορεί να αλλάξει π.χ το χειριστή διακοπών ρολογιού – προστατεύεται μέσω του υλικού).
- Πολλά τμήματα που φαινομενικά αποτελούν το ΛΣ (π.χ. διεπαφή shell, GUI) βρίσκονται στο user space.

1.1 Τι είναι το Λειτουργικό Σύστημα (ΛΣ)

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Συστατικά Η/Υ

- Ένας σύγχρονος Η/Υ αποτελείται από:
  - Έναν ή περισσότερους επεξεργαστές
  - Κύρια μνήμη
  - Δίσκους
  - Εκτυπωτές
  - Διάφορες συσκευές εισόδου/εξόδου (Ε/Ε)
- Η διαχείριση αυτών των συστατικών, απαιτεί την ύπαρξη ενός στρώματος λογισμικού – του *Λειτουργικού Συστήματος*

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- Το Λ.Σ. διαχειρίζεται όλες τις λεπτομέρειες που αφορούν τα συστατικά (το υλικό) ενός υπολογιστικού συστήματος.
- Για αυτό το λόγο είναι ιδιαίτερα σύνθετα, λεπτομερή και μεγάλα προγράμματα
  - (περίπου 5.000.000 γραμμές κώδικα ή 100 τόμοι των 1000 σελίδων)
  - Ο κώδικας του GUI είναι ×10 ή ×20 !
  - Αλλάζουν δύσκολα

Οι δύο όψεις των ΛΣ (1/2)

- (A) Μία εκτεταμένη μηχανή (extended machine)
  - Η αρχιτεκτονική των Η/Υ είναι ιδιαίτερα σύνθετο θέμα (instruction set, διαχείριση μνήμης, Ε/Ε).
  - Το ΛΣ **αποκρύπτει τις σύνθετες λεπτομέρειες** που εκτελεί ένας Η/Υ.
  - Παρουσιάζει στον χρήστη μία ***«εικονική μηχανή»*** (virtual machine), πολύ πιο απλή στη χρήση της.
  - Χρησιμοποιεί την έννοια της ***αφαίρεσης (abstraction)***
  - Προσέγγιση top-down (από την κορυφή προς τη βάση).

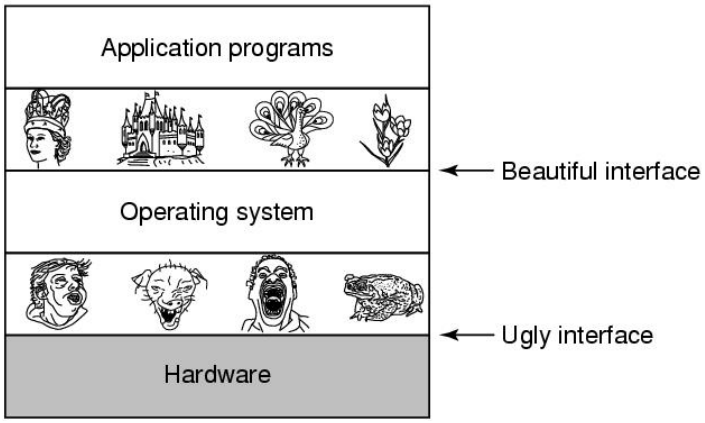
Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

(A) Επεκτεταμένη μηχανή

•Ακόμη και το πιο απλό περιφερειακό (μία δισκέτα) έχει έναν ελεγκτή (controller) ο οποίος θα πρέπει να ελέγξει:

- Ένα σύνολο εντολών (instructions) – η απλή δισκέτα π.χ. έχει 16 εντολές
- Φόρτωση αυτών σε καταχωρητές εντός της συσκευής (device register)
- Διαταγές ανάγνωσης, εγγραφής, διαμόρφωσης δίσκου, προετοιμασία, ανίχνευση θέσης, επανεκκίνηση κτλ
- Πάρα πολλές λεπτομέρειες θα πρέπει να καθοριστούν για μία απλή ανάγνωση
- Ο προγραμματιστής δεν θέλει να ασχολείται με όλες αυτές τις εξαντλητικές λεπτομέρειες.
- Είναι προτιμότερο να βλέπει κάποια απλή και υψηλότερου επιπέδου αφαίρεση (abstraction), π.χ. αρχείο δεδομένων με κάποιο όνομα.

(Α) Το ΛΣ ως μία εκτεταμένη μηχανή



Εικόνα 1-2. Το ΛΣ μετασχηματίζει το «άσχημο υλικό» σε «όμορφες αφαιρέσεις» / διεπαφές χρήστη.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- Το Λ.Σ. κρύβει το «άσχημο υλικό» από τον χρήστη (προγραμματιστή) και δίνει όμορφες, σαφείς και συναφείς αφαιρέσεις μέσα από προγραμματιστικές διεπαφές (interfaces).

Οι δύο όψεις των ΛΣ (2/2)

- (Β) Ένας διαχειριστής πόρων
  - Προσέγγιση bottom-up (από τη βάση προς την κορυφή)
  - Το ΛΣ υπάρχει για να διαχειρίζεται τους πόρους του Η/Υ, (CPU, μνήμη, δίσκο κτλ).
  - Ρόλος του ΛΣ: Η (δίκαιη) ***κατανομή των πόρων (resource allocation)***
    - Τα προγράμματα των χρηστών **ανταγωνίζονται** να χρησιμοποιήσουν τους πόρους αυτούς.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

(Β) Διαχειριστής πόρων

•Παράδειγμα: Τι θα γινόταν εάν πολλά προγράμματα στέλνονταν στον εκτυπωτή ταυτόχρονα;

- Ενδέχεται να τυπωθούν μερικές γραμμές από το ένα πρόγραμμα, μετά μερικές από το δεύτερο κ.ο.κ.
- Χρειάζεται έλεγχος και διαμοίραση του πόρου εκτυπωτής.

(Β) Το ΛΣ ως ένας διαχειριστής πόρων

- Επιτρέπει την ταυτόχρονη εκτέλεση πολλών προγραμμάτων χρησιμοποιώντας **πολύπλεξη πόρων (multiplexing)** με δύο διαφορετικούς τρόπους:
  1. **Χρονική πολύπλεξη** (time multiplexing)
  2. **Χωρική πολύπλεξη** (space multiplexing)
- Η πολύπλεξη δημιουργεί θέματα ασφάλειας, προστασίας, αμεροληψίας κτλ.
  - Το ΛΣ διαχειρίζεται και προστατεύει τη μνήμη, τις συσκευές Εισόδου/Εξόδου (Ε/Ε) και άλλους πόρους.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

1. Παράδειγμα **χρονικής πολύπλεξης**: Κάθε πρόγραμμα θέλει να χρησιμοποιήσει για κάποιο χρονικό διάστημα ένα ή περισσότερους διαμοιραζόμενους πόρους (π.χ. επεξεργαστή - CPU)
2. Παράδειγμα **χωρικής πολύπλεξης**: Κάθε πρόγραμμα θέλει να χρησιμοποιήσει κάποιο χώρο από ένα ή περισσότερους διαμοιραζόμενους πόρους (π.χ. ταυτόχρονη φόρτωση 2 προγραμμάτων σε δύο περιοχές της κύριας μνήμης. Ή κοινή χρήση σκληρού δίσκου)

1.2 Η ιστορία των ΛΣ

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Ιστορία των ΛΣ

- **1η γενιά 1945 - 1955**
  - Λυχνίες κενού, καλωδίωση συνδέσεων
- **2η γενιά 1955 - 1965**
  - Κυκλώματα (transistors), Συστήματα δέσμης (batch systems)
- **3η γενιά 1965 – 1980**
  - Ολοκληρωμένα κυκλώματα (Integrated Circuits) και πολύ-προγραμματισμός (multiprogramming)
- **4η γενιά 1980 – σήμερα**
  - Προσωπικοί υπολογιστές

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

1η γενιά

•Υπολογιστές όπως (Z3, Collosus, ENIAC)

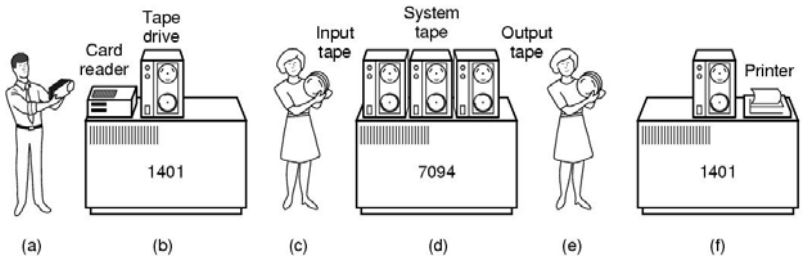
•Ο προγραμματισμός γινόταν με καλωδίωση του πίνακα συνδέσεων (plug boards) ή σε απόλυτη γλώσσα μηχανής

•Μόνο για απλές αριθμητικές πράξεις

•Πολύ χρόνο (αν στο ενδιάμεσο δεν είχε καεί κάποια λυχνία)

•Τα ΛΣ δεν υπάρχουν

2η γενιά
Κυκλώματα και Συστήματα δέσμης (1)

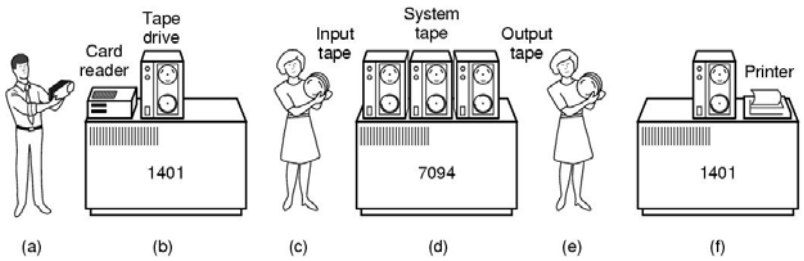


Εικόνα 1-3. Ένα πρώιμο σύστημα δέσμης.
(a) Ο προγραμματιστής φέρνει την κάρτα με το πρόγραμμα στο 1401.
(b)Το 1401 περνάει τις δέσμες εργασιών στην ταινία εισόδου.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

1. Τα transistor μείωσαν το κόστος και αύξησαν τη χρησιμότητα των υπολογιστών, παρόλα αυτά ήταν δυνατή η χρήση τους μόνο από μεγάλες επιχειρήσεις / οργανισμούς.
2. Mainframes και εργασίες δέσμης (jobs): Αρχικά οι προγραμματιστές έγγραφε το πρόγραμμα, το «τρύπαγε» σε διάτρητες κάρτες εισόδου, το παρέδιδε στο χειριστή και περίμενε τα αποτελέσματα. Αργότερα πέρναγε να πάρει τα αποτελέσματα. Πολύς (και ακριβός) υπολογιστικός χρόνος πήγαινε χαμένος.
3. Λύση: τα **συστήματα δέσμης (batch systems)**: Χρήση υπολογιστών χαμηλών επιδόσεων για Ε/Ε και υψηλής επίδοσης για πραγματικό υπολογισμό.

2η γενιά
Κυκλώματα και Συστήματα δέσμης(2)

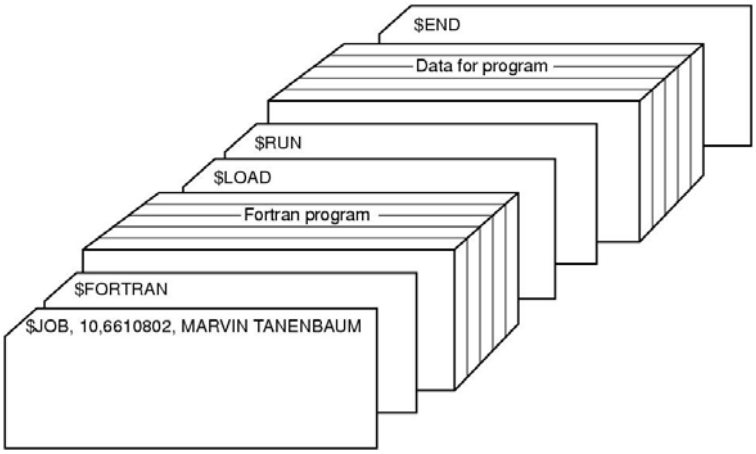


Εικόνα 1-3. (c) Ο χειριστής φέρνει την ταινία εισόδου στο 7094.
(d) Το 7094 κάνει τους υπολογισμούς.
(e) Ο χειριστής φέρνει την ταινία εξόδου στο 1401.
(f) Το 1401 τυπώνει τα αποτελέσματα εξόδου.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- Το πρόγραμμα που χειριζόταν την ανάγνωση κάθε ταινίας εισόδου στο σύστημα επεξεργασίας και αναλάμβανε την αλλαγή του ήταν ο **πρόγονος των σύγχρονων ΛΣ**.

2η γενιά
Κυκλώματα και Συστήματα δέσμης(3)

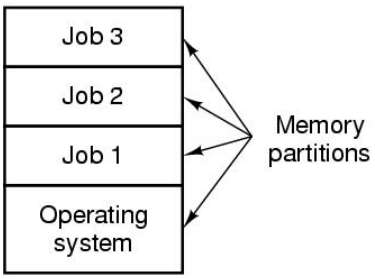


Εικόνα 1-4. Δομή μιας τυπικής εργασίας FMS (Fortran Monitor System).

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- Το ΛΣ της εποχής ήταν το FMS. Η δομή της τυπικής εργασίας εισόδου είναι η εξής:
  - Η 1η κάρτα έδειχνε την εκκίνηση μίας εργασίας, το χρόνο εκτέλεσης, τον προγραμματιστή κτλ.
  - Η 2η κάρτα ($FORTRAN) έδειχνε στο ΛΣ να φορτώσει το μεταγλωττιστή της γλώσσας από την ταινία συστήματος.
  - Στη συνέχεια υπήρχαν κάρτες με τον μεταγλωττισμένο κώδικα (object code).
  - Η $LOAD υποδείκνυε στο ΛΣ να φορτώσει τον κώδικα και η $RUN να τον τρέξει.
  - …
  - Αυτές οι κάρτες ελέγχου είναι οι πρόγονοι των σύγχρονων **γραμμών εντολών** (command shells).
  - Χρήσεις υπολογιστών: επιστημονικοί και τεχνικοί υπολογισμοί.
  - Τυπικά ΛΣ της εποχής: FMS, IBSYS

3η γενιά: Ολοκληρωμένα κυκλώματα και πολυπρογραμματισμός (1/2)



Εικόνα 1-5. Ένα σύστημα πολύπρογραμματισμού (multiprogramming) με τρεις εργασίες στη μνήμη.

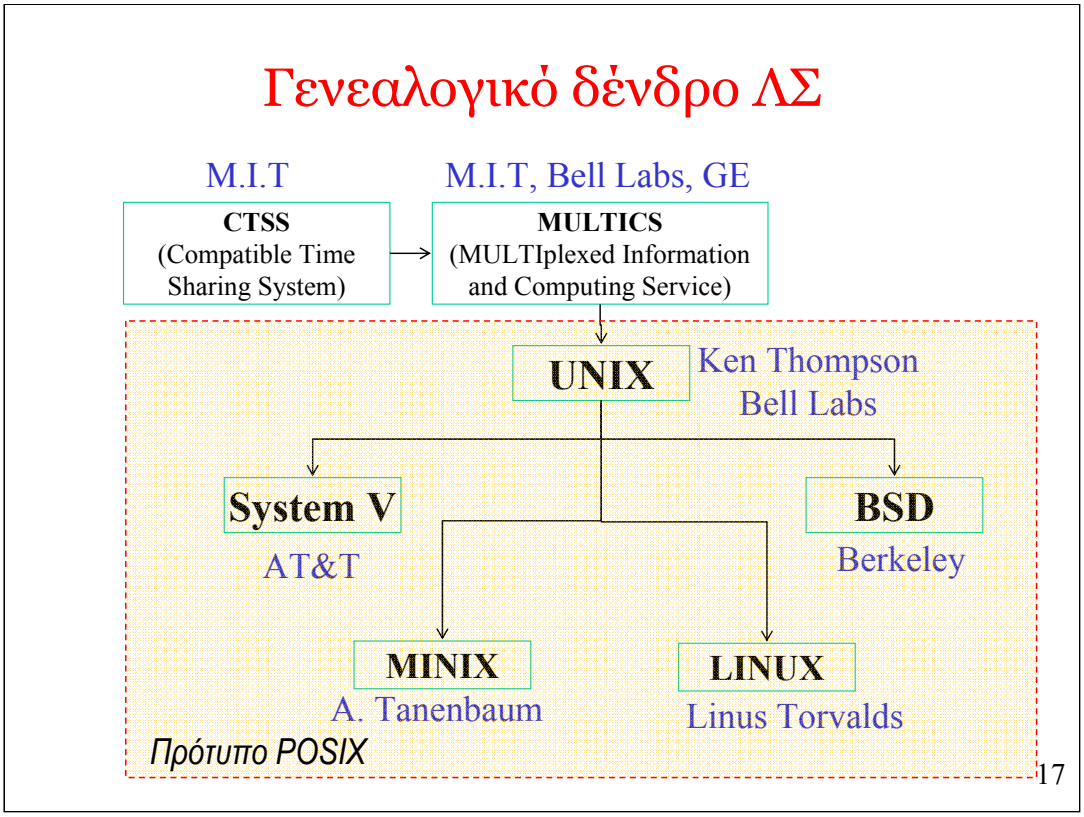
Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

1. Στις αρχές της 3η γενιάς υπήρχαν 2 γραμμές παραγωγής υπολογιστών: επιστημονικοί (IBM 7094) και για εργασίες προσανατολισμένες σε Ε/Ε (IBM 1401) π.χ. ταξινόμηση.
2. Η IBM παρουσίασε για πρώτη φορά οικογένειες **συμβατών υπολογιστών**: έτρεχαν το ίδιο λογισμικό, απλά είχαν διαφορετικές δυνατότητες (σειρά System/360 – 370, …, zSeries).
3. Η σειρά 360 ήταν η πρώτη που χρησιμοποιούσε **ολοκληρωμένα κυκλώματα** – άρα καλύτερη τιμή/απόδοση.
4. Πρόβλημα με το κοινό λογισμικό: πώς θα συνδυαστούν διαφορετικές ανάγκες συστημάτων στο ίδιο λογισμικό;
    - Αποτέλεσμα: πολύπλοκο ΛΣ πολύ μεγαλύτερο από το FMS, με πολλά-πολλά σφάλματα.
5. Εισήγαγε όμως την έννοια του **πολυπρογραμματισμού**.
    - Καλύτερη χρησιμοποίηση της CPU: όταν μία εργασία περιμένει Ε/Ε, κάποια άλλη χρησιμοποιεί τη CPU.
    - Είχε υλικό προστασίας, για την αποφυγή παρεμβάσεων μίας εργασίας στο χώρο μνήμης μίας άλλης.

3η γενιά: Ολοκληρωμένα κυκλώματα και πολυπρογραμματισμός (2/2)

- Άλλες καινοτομίες των ΛΣ 3ης γενιάς
  - Παροχέτευση (spooling): Οι εργασίες διαβάζονται από το δίσκο (πολλές στη σειρά). Με την ολοκλήρωση μίας εργασίας, φόρτωση της επόμενης σε ένα ελεύθερο τμήμα της μνήμης.
  - Πρόβλημα: μεγάλος χρόνο απόκρισης (λόγω της σειριακής εκτέλεσης)
  - Χρονομερισμός (time-sharing):
    - Κάθε χρήστης έχει για ένα μικρό διάστημα τη CPU.
    - Προτεραιότητα στις αλληλεπιδραστικές εργασίες των χρηστών. Οι βαριές εργασίες δέσμης τρέχουν στο παρασκήνιο, όταν η CPU είναι ελεύθερη.

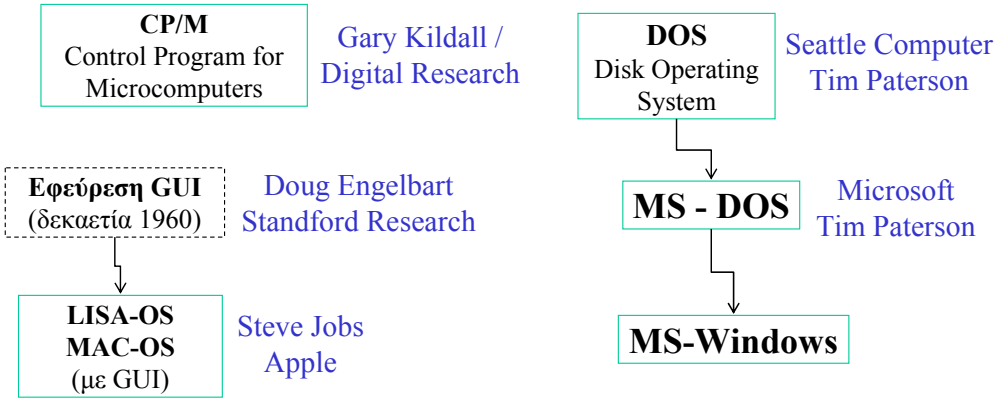
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**



1. CCTS: Εισήγαγε το χρονομερισμό. Μικρή επιτυχία.
2. MULTICS:
   - Ιδέα για έναν υπολογιστή-πάροχο, για 100αδες χρήστες με χρονομερισμό.
   - Πολύ φιλόδοξο, μικρή επιτυχία.
   - Χρησιμοποιήθηκε μέχρι το ΄90.
   - Μεγάλη επιρροή στα ΛΣ.
3. UNIX: Ξεκίνησε ως απλοποιημένη έκδοση του MULTICS για 1 χρήστη.
   - Ο πηγαίος κώδικας ήταν διαθέσιμος οπότε οδήγησε σε πολλές (μη συμβατές) εκδόσεις.
4. POSIX:
   - για να είναι δυνατή η συγγραφή προγραμμάτων για UNIX χωρίς προβλήματα συμβατότητας, το IEEE έφτιαξε το πρότυπο αυτό.
   - Υποστηρίζεται από τις σύγχρονες εκδόσεις.
   - Ουσιαστικά περιλαμβάνει ένα ελάχιστο σύνολο διασυνδέσεων κλήσεων συστήματος (system call interface) που θα πρέπει να υποστηρίζουν όλα τα UNIX ΛΣ.

4η γενιά: Προσωπικοί υπολογιστές (2/3)

Η ανάπτυξη κυκλωμάτων LSI (Large Scale Integation) οδήγησε στην δημιουργία των μικροϋπολογιστών



Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

CP/M (1974)

1. Η Intel δημιουργεί τον επεξεργαστή 8080 (8 bit) και αναζητά δοκιμαστικό ΛΣ.
2. Ο Kildall φτιάχνει ένα ελεγκτή δίσκου (controller) για οδηγό δισκέτας, δημιουργώντας τον 1ο μικροϋπολογιστή με δισκέττα.
3. Η Intel δεν πιστεύει στους μικροϋπολογιστές με δισκέττα, ο Kildall ιδρύει δική του εταιρεία (Digital Research)

DOS

1. Στις αρχές του ‘80 η IBM ψάχνει ΛΣ για το IBM PC. Επικοινωνεί με τον Bill Gates για να αποκτήσει την άδεια χρήσης του διερμηνευτή BASIC.
2. Ο Gates τους συνέστησε την Digital Research για ανάπτυξη ΛΣ για το IBM PC.
3. Ο Kildal αρνήθηκε να υπογράψει μυστική συμφωνία με την IBM για το PC. Η IBM επέστρεψε στον Gates.
4. Ο Gates αγοράζει το DOS από μία μικρή εταιρεία και προσφέρει το πακέτο DOS/BASIC στην IBM. Το πακέτο πουλιέται μαζί με τον υπολογιστή και κάνει τεράστια εμπορική επιτυχία.

Apple

1. Τη δεκαετία του ‘60 στο Standford Research Institute ο Dough Engelbart ανακαλύπτει το Graphical User Interface (GUI). Οι ιδέες ενσωματώθηκαν από ερευνητές στο XEROX PARC.
2. Ο Steve Jobs επισκέπτεται το XEROX PARC και αμέσως κατάλαβε την αξία του GUI. Κατασκευάζει στο γκαράζ του τον πρώτο του υπολογιστή LISA με GUI. Ήταν αρκετά ακριβός και δεν έκανε επιτυχία.
3. Η δεύτερη προσπάθεια Apple Macintosh έκανε τεράστια επιτυχία.

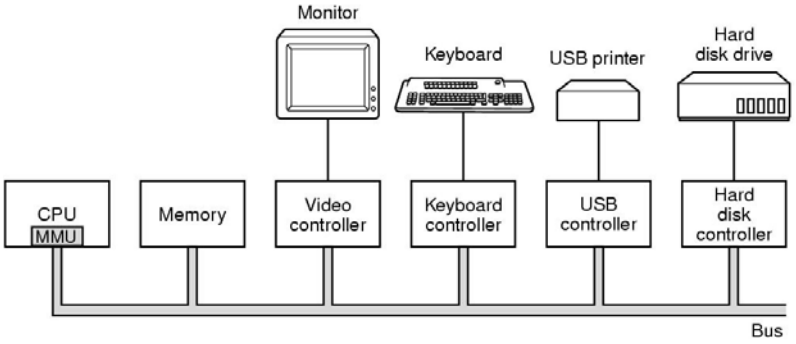
Windows:

1. (1985-1995) Η microsoft δημιουργεί ένα ΛΣ με GUI, το MS-Windows. Στην πραγματικότητα εκκινεί με DOS και μετά φορτώνει το κέλυφος με το παραθυρικό περιβάλλον.

1.3 Το υλικό των Η/Υ

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Το υλικό ενός σύγχρονου Η/Υ



Εικόνα 1-6. Μια απλή αναπαράσταση των συστατικών ενός βασικού συστήματος Η/Υ.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

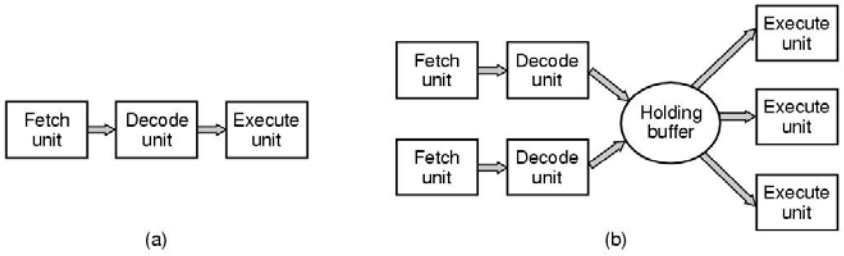
Κεντρική μονάδα επεξεργασίας (CPU)

- Κύκλος λειτουργίας CPU: μεταφορά της επόμενης εντολής από τη μνήμη, την αποκωδικοποίηση του τύπου της, και την εκτέλεση.
- Η μεταφορά διαρκεί περισσότερο, γι' αυτό χρησιμοποιεί καταχωρητές (**registers**):
  1. Μετρητής προγράμματος (program counter): δείχνει τη διεύθυνση της επόμενης εντολής.
  2. Δείκτης στοίβας (stack pointer): δείχνει την κορυφή της τρέχουσας στοίβας στη μνήμη.
  3. Πλαίσιο στοίβας (stack frame): έχει τις παραμέτρους εισόδου, τοπικές και προσωρινές μεταβλητές που δεν διατηρούνται σε καταχωρητές.
  4. (Program Status Word) – PSW: δείχνει την κατάσταση λειτουργίας (πυρήνα ή χρήστη), την προτεραιότητα στη CPU, bit ελέγχου κτλ.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Το ΛΣ πρέπει να έχει γνώση όλων αυτών των παραμέτρων, για κάθε πρόγραμμα σε περίπτωση πολύπλεξης.

Διοχέτευση επεξεργαστή (CPU Pipelining)



Εικόνα1-7. (a) Μία σωλήνωση τριών σταδίων. (b) Υπερβαθμωτή CPU.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

(α) Για καλύτερη απόδοση, το απλό μοντέλο προσκόμισης – αποκωδικοποίησης – εκτέλεσης μίας εντολής ανά φορά έχει αλλάξει.

- Η CPU έχει διαφορετικές μονάδες προσκόμισης, αποκωδικοποίησης και εκτέλεσης εντολών – ονομάζεται διοχέτευση επεξεργαστή (pipelining).
- Μπορεί ταυτόχρονα να προσκομίζει μία εντολή, να αποκωδικοποιεί μία δεύτερη και να εκτελεί μία τρίτη.

(β) Στην υπερβαθμωτή CPU υπάρχουν πολλαπλές μονάδες fetch, decode, execute. Π.χ. μία για αριθμητική ακεραίων, άλλη για κινητής υποδιαστολής άλλη για Boolean πράξεις.

- Για την εκτέλεση των πολλαπλών εντολών χρησιμοποιείται buffer.

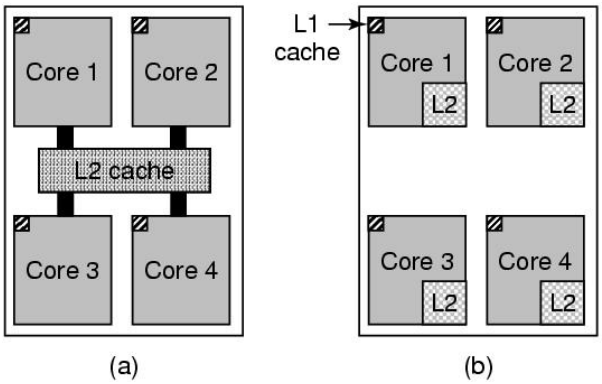
CPU και κατάσταση πυρήνα

- Η CPU έχει **κατάσταση πυρήνα** και **κατάσταση χρήστη.** Η κατάσταση εκτέλεσης κάθε εντολής ελέγχεται από bit ελέγχου στον καταχωρητή PSW.
- Στην κατάσταση πυρήνα εκτελούνται όλες οι εντολές, σε κατάσταση χρήστη μόνο ορισμένες.
- Το ΛΣ εκτελείται σε κατάσταση πυρήνα.
- Τα προγράμματα χρήστη που θέλουν ειδικές λειτουργίες πραγματοποιούν **κλήσεις συστήματος (system calls)**, η οποία προκαλεί **παγίδευση (trap)** και μεταβαίνει σε κατάσταση πυρήνα.
- Η εντολή TRAP προκαλεί την ανάμιξη του ΛΣ.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- Η εντολή TRAP προκαλεί αλλαγή από κατάσταση χρήστη σε κατάσταση πυρήνα και ξεκινά την ανάμιξη του ΛΣ. Όταν η εργασία ολοκληρωθεί, ο έλεγχος επιστρέφει στο πρόγραμμα χρήστη, στην εντολή που ακολουθεί την system call.
- Περισσότερα στη συνέχεια, προς το παρόν θεωρείστε την TRAP ως μία ειδική εντολή.
- Το υλικό έχει και άλλες παγιδεύσεις για προστασία από κάποια λάθος κατάσταση (π.χ. διαίρεση με το 0, underfow υποδιαστολής κτλ)

Επεξεργαστές πολλαπλών νημάτων (Multithread) και πολλαπλών πυρήνων (Multicore)

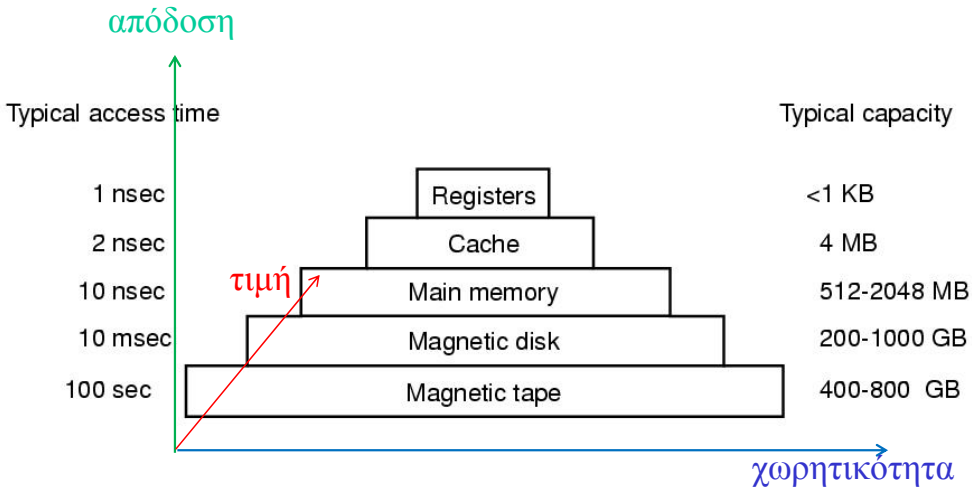


Εικόνα 1-8. (a) Chip τεσσάρων πυρήνων με διαμοιραζόμενη μνήμη cache L2.
(b) Chip τεσσάρων πυρήνων με ξεχωριστή μνήμη cache L2.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- Νόμος του Moore: Υπολογιστική ισχύς 2πλασιάζεται κάθε 18 μήνες.
- Οι Cache (κρυφές) μνήμες, αυξάνουν την απόδοση των (πολύ) πιο γρήγορων επεξεργαστών, αλλά έχουν κόστος.
- Υπερνημάτωση ή πολυνημάτωση (multithreading). Αυξάνει την απόδοση αλλά κάθε επεξεργαστής τρέχει ένα νήμα κάθε φορά.
- Πολλαπλοί **πυρήνες** (dual core, multi core): Περισσότεροι πραγματικοί πλήρεις επεξεργαστές
- Κρυφή μνήμη επιπέδου 1 (**L1-cache)**: πολύ γρήγορη αλλά και ακριβότερη.
- Κρυφή μνήμη επιπέδου 2 (**L2-cache)**: λιγότερο γρήγορη από την 1 αλλά και λιγότερο ακριβότερη.

Μνήμη
(Ιεραρχία)



Εικόνα 1-9. Μία τυπική ιεραρχία της μνήμης.
Οι αριθμοί αποτελούν πολύ γενικές προσεγγίσεις.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

- Επιθυμητά χαρακτηριστικά: μεγάλη ταχύτητα, μεγάλη χωρητικότητα, χαμηλό κόστος.
- Δεν υπάρχει κάποια μνήμη που να ικανοποιεί όλα τα χαρακτηριστικά, και γι' αυτό το σύστημα μνήμης είναι οργανωμένο σε μία ιεραρχία επιπέδων.

Μνήμη
(Καταχωρητές)

1. Καταχωρητές (registers)

- Κορυφαίο επίπεδο μνήμης, πολύ υψηλό κόστος.
- Βρίσκονται στο εσωτερικό της CPU, άρα προκαλούν ελάχιστες καθυστερήσεις.
- Μέγεθος: (< 1KB) π.χ.,
  - 32 × 32 bit ή 64 × 64 bit.
- Τα προγράμματα αποφασίζουν ποια δεδομένα θα διατηρούνται εκεί.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Μνήμη
(Κρυφή μνήμη)

2. Κρυφή μνήμη (Cache):

- Δεύτερο επίπεδο, ελέγχεται από το υλικό.
- Χωρίζεται σε **γραμμές κρυφής μνήμης (cache lines)** των 64 byte.
- Όσες γραμμές χρησιμοποιούνται συχνότερα, διατηρούνται στην ταχύτερη κρυφή μνήμη που βρίσκεται **μέσα στην CPU.**
- Όταν το πρόγραμμα βρίσκει τα δεδομένα που ψάχνει μέσα στην μνήμη αυτή της CPU, τότε έχουμε ***ευστοχία μνήμης (cache hit).***
- Οι ευστοχίες διαρκούν κατά Μ.Ο. 2 κύκλους ρολογιού, μετά γίνεται αναζήτηση της ζητούμενης λέξης στη μνήμη με μεγάλη καθυστέρηση.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Μνήμη
(κρυφή μνήμη -επίπεδα)

Επίπεδα κρυφής μνήμης (cache memory layers):

L1 cache memory (συνήθως 16 KB):

- Βρίσκεται πάντα στο εσωτερικό της CPU και τροφοδοτεί τη λογική μονάδα εκτέλεσης με αποκωδικοποιημένες εντολές.
- Πολλά chip έχουν και δεύτερη L1 μνήμη για τις συχνότερα χρησιμοποιούμενες λέξεις δεδομένων.

L2 cache memory (μερικά MB)

- Υπάρχει συχνά, είναι αρκετά μεγαλύτερη και διατηρεί τις πρόσφατα χρησιμοποιούμενες από τη CPU λέξεις.
- Η L2 έχει μεγαλύτερη καθυστέρηση από την L1 (π.χ. 1-2 κύκλους ρολογιού).

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Μνήμη
(κρυφή μνήμη -επίπεδα)

Ερωτήσεις σχετικά με την κρυφή μνήμη (cache memory):

- Πότε να μπει ένα νέο τμήμα στην μνήμη cache.
- Σε ποια θέση της μνήμης cache θα μπει.
- Ποιο τμήμα θα αφαιρεθεί πρώτα από την μνήμη cache, όταν υπάρξει ανάγκη για κενή θέση.
- Σε ποια μεγαλύτερη μνήμη να τοποθετηθεί ένα τμήμα που μόλις αφαιρέθηκε από την cache.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

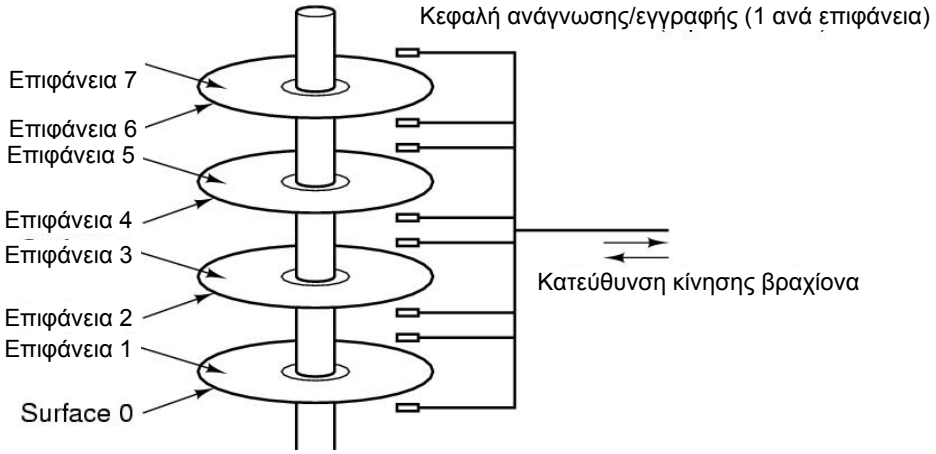
Μνήμη
(Κύρια Μνήμη)

3. Μνήμη Τυχαίας Προσπέλασης (Random Access Memory – RAM)

- Ο εργάτης του συστήματος.
- Όταν μία αίτηση της CPU δεν μπορεί να ικανοποιηθεί από την κρυφή μνήμη, τότε μεταφέρεται στην κύρια μνήμη.
- Η RAM είναι πτητική μνήμη (τα δεδομένα χάνονται όταν κλείσει το ρεύμα).
- Η **μνήμη ROM (Read Only Memory) είναι μη πτητική.**
  - Διατηρεί πληροφορία όπως τον boot loader.
- Η μνήμες EEPROM (Electrically Erasable Programmable ROM) μπορούν να ξαναγραφούν.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Δίσκοι



Εικόνα 1-10. Δομή ενός οδηγού δίσκου.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

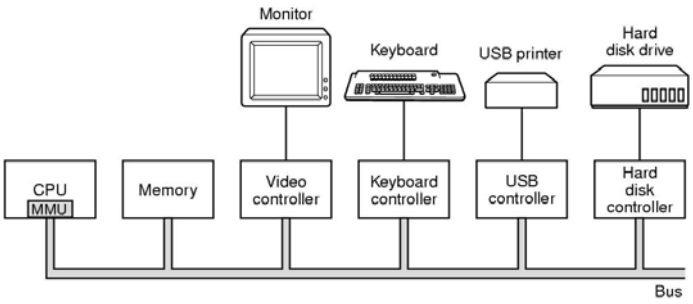
- Η αποθήκευση στο σκληρό δίσκο είναι 2 τάξεις μεγέθους (100 φορές) φθηνότερη ανά bit από την RAM.
- …αλλά και 3 τάξεις μεγέθους (1000 φορές!) πιο αργή επειδή είναι **μηχανική συσκευή.**
- Αποτελείται από πολλές μεταλλικές πλάκες που περιστρέφονται με ταχύτητα 5400, 7200 ή 10800 στροφές ανά λεπτό (rpm).
- Ένα μεταλλικό βραχίονα που κινείται πάνω στις πλάκες.
- Τα δεδομένα γράφονται πάνω στο δίσκο φτιάχνοντας σειρές από ομόκεντρους κύκλους.
    - **Τροχιά (track)**: δακτύλιος με κάποια ακτίνα πάνω στην επιφάνεια.
    - **Κύλινδρος (cylinder)**: το σύνολο των τροχιών μίας δεδομένης ακτίνας πάνω σε όλες τις επιφάνειες.
    - **Τομέας (sector)**: οι τροχιές διαιρούνται σε τομείς, δηλαδή περιοχές δεδομένου μήκους (π.χ. 512 byte)
- Η μετακίνηση του βραχίονα σε έναν κύλινδρο διαρκεί π.χ. 5-10 msec.
- Η μετακίνηση του ζητούμενου τομέα κάτω από την κεφαλή, προσθέτει άλλα 5-10 msec.
- Χρόνος ανάγνωσης ή εγγραφής: περίπου 50 – 160 MB/sec.
- Η **εικονική μνήμη** δίνει τη δυνατότητα να εκτελεστούν προγράμματα που έχουν μέγεθος μεγαλύτερο από τη φυσική μνήμη, χρησιμοποιώντας το δίσκο σαν να ήταν κύρια μνήμη.
    - Χρησιμοποιεί τη μονάδα διαχείρισης μνήμης (MMU) της CPU.

Συσκευές Εισόδου/Εξόδου (Ε/Ε)
(Ελεγκτής συσκευής)

Ελεγκτής συσκευής (controller): chip για το φυσικό έλεγχο της συσκευής.

•Ο ελεγκτής παρουσιάζει στο ΛΣ μία λιγότερο (αλλά και πάλι αρκετά) σύνθετη εικόνα.

•Το ΛΣ βλέπει **μόνο τη διασύνδεση με τον ελεγκτή**, και όχι τη διασύνδεση μεταξύ του ελεγκτή και της φυσικής συσκευής. (Π.χ. IDE controller, SCSI controller κ.ο.κ.)



Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

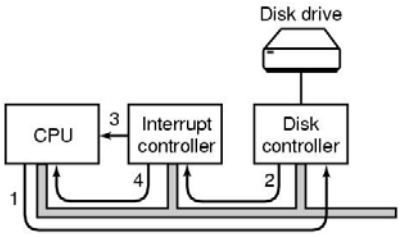
Συσκευές Εισόδου/Εξόδου (Ε/Ε)
(Οδηγός συσκευής)

- Η διασύνδεση του ΛΣ με τον ελεγκτή γίνεται μέσα από τον **οδηγό συσκευής (device driver)**.
- Τοποθέτηση driver στον πυρήνα του ΛΣ: (3 μέθοδοι)
  1. Επανασύνδεση του πυρήνα με το νέο οδηγό και επανεκκίνηση (UNIX).
  2. Δημιουργία μίας καταχώρησης για τον οδηγό της συσκευής σε κάποιο ειδικό αρχείο και επανεκκίνηση (Windows).
  3. Δυναμική φόρτωση (on-the-fly).
- Κάθε ελεγκτής συσκευής έχει έναν αριθμό καταχωρητών για να επικοινωνεί με τον έξω κόσμο.
  - Το σύνολο των καταχωρητών μίας συσκευής Ε/Ε αποτελεί το **χώρο θυρών Ε/Ε (I/O port space)**

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- Ο κατασκευαστής κάθε συσκευής Ε/Ε παρέχει έναν driver για κάθε ΛΣ.

Συσκευές Εισόδου Εξόδου (Ε/Ε)
(διακοπές – interrupts)



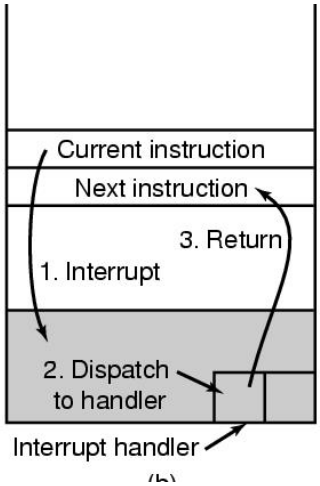
1. Ο οδηγός συσκευής δίνει οδηγίες στον ελεγκτή δίσκου (γράφοντας στους καταχωρητές του ελεγκτή). Ο ελεγκτής ξεκινά τη συσκευή.
2. Μόλις τελειώσει η ανάγνωση/ εγγραφή ο ελεγκτής στέλνει σήμα στον ελεγκτή διακοπών.
3. Ο ελεγκτής διακοπών ενεργοποιεί τη διακοπή στη CPU.
4. Τοποθέτηση αριθμού συσκευής στο δίαυλο.

Εικόνα 1-11. (α) Τα βήματα εκκίνησης μίας συσκευής Ε/Ε και η λήψη της διακοπής (interrupt).

- Στο βήμα 4, ο αριθμός συσκευής τοποθετείται στο δίαυλο ώστε η CPU να μάθει ποια συσκευή τελείωσε την εργασία της (πολλές συσκευές μπορεί ταυτόχρονα να είναι σε λειτουργία).

Συσκευές Εισόδου Εξόδου (Ε/Ε)

(Χειριστής διακοπών)



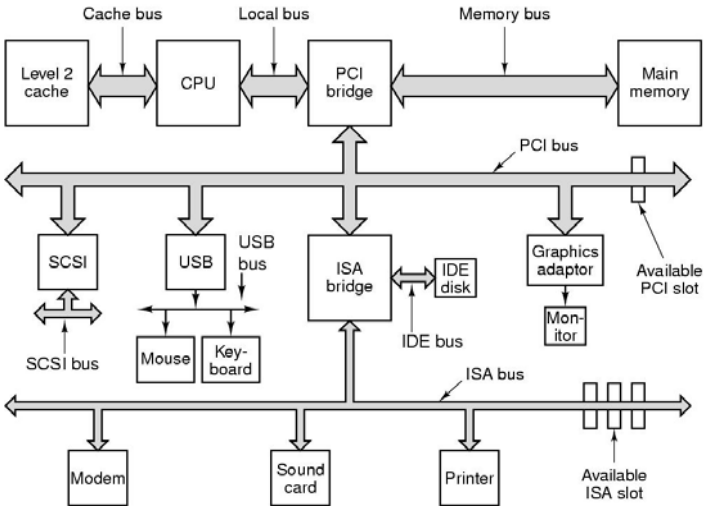
1. Γίνεται αποδεκτή η διακοπή.
   - Ο μετρητής προγράμματος (program counter) και ο PSW τοποθετούνται στη στοίβα.
   - Μετάβαση της CPU σε κατάσταση πυρήνα. Αυτό το τμήμα της μνήμης λέγεται διάνυσμα διακοπών (interrupt vector).
2. Ο χειριστής διακοπών αφαιρεί το μετρητή προγράμματος και τον PSW από τη στοίβα και τους αποθηκεύει. Διερεύνηση της συσκευής και Ε/Ε.
3. Ο χειριστής διακοπών ολοκληρώνει την εργασία του και επιστρέφει τον έλεγχο στο πρόγραμμα χρήστη.

Εικόνα 1-11. (β) Η επεξεργασία των διακοπών της συσκευής.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

- Εφόσον η CPU δεχθεί τη διακοπή, ο μετρητής προγράμματος (program counter) και ο PSW τοποθετούνται στην τρέχουσα στοίβα και γίνεται μετάβαση της CPU σε κατάσταση πυρήνα.
- Όταν ο χειριστής ολοκληρώσει την εργασία του, επιστρέφει τον έλεγχο στο πρόγραμμα χρήστη που εκτελούνταν προηγουμένως, στην 1η εντολή μετά τη διακοπή.

Δίαυλοι (Buses)



Εικόνα 1-12. Η δομή ενός συστήματος Pentium

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

- Πλέον ένας δίαυλος δεν αρκεί, για το λόγο αυτό υπάρχουν σε ένα σύγχρονο σύστημα πολλοί ειδικοί δίαυλοι.
- Δίαυλοι:
  - ISA (Industry Standard Architecture): για ιστορικούς λόγους (8.33 MHz, 2 byte ταυτόχρονα, ταχύτητα 16.67 MB/sec)
  - PCI (Peripheral Component Interconnect): (66MHz, 8byte ταυτόχρονα, ταχύτητα 528 MB/sec)
  - PCI Express
- Εξειδικευμένοι δίαυλοι:
  - SCSI (160 MB/sec)
  - IDE
  - USB (1.0 – 1.5 MB/sec, 2.0 60 MB/sec).

Εκκίνηση του Η/Υ

1. Το πρόγραμμα BIOS ελέγχει τις συσκευές, τη RAM, τους διαύλους, τις συσκευές άμεσης λειτουργίας.
   - Το BIOS (Basic Input Output System) βρίσκεται επάνω στο motherboard και υποστηρίζει λειτουργίες Ε/Ε χαμηλού επιπέδου.
2. Το BIOS αποφασίζει ποια είναι η συσκευή εκκίνησης (σκληρός δίσκος, CD, flash κτλ).
3. Από τον τομέα εκκίνησης (boot sector) της συσκευής εκκίνησης, βρίσκεται ποιο είναι το ενεργό partition.
4. Φορτώνεται ο boot loader και από εκεί το ΛΣ του ενεργού partition.
5. Το ΛΣ βρίσκει από το BIOS τους οδηγούς των συσκευών και τους φορτώνει στον πυρήνα.
6. Το ΛΣ δίνει αρχικές τιμές στους πίνακές του, δημιουργεί τις απαραίτητες διεργασίες παρασκηνίου και ξεκινά το πρόγραμμα login.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

1.4 Οι κατηγορίες των Λ.Σ.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Ο «ζωολογικός κήπος» των ΛΣ

1. ΛΣ μεγάλων υπολογιστών (Mainframe OS)
2. ΛΣ εξυπηρετητών (Server OS)
3. ΛΣ πολυεπεξεργαστικών συστημάτων (Multiprocessor OS)
4. ΛΣ προσωπικών υπολογιστών (Personal computer OS)
5. ΛΣ υπολογιστών χειρός (Handheld OS)
6. ΛΣ ενσωματωμένων συστημάτων (Embedded OS)
7. ΛΣ αισθητήρων (Sensor node OS)
8. ΛΣ πραγματικού χρόνου (Real-time OS)
9. ΛΣ έξυπνων καρτών (Smart card OS)

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

ΛΣ μεγάλων υπολογιστών
(Mainframes)

- Πολύ μεγάλοι υπολογιστές
- Βασική διαφορά: η δυνατότητες Ε/Ε
- Προσανατολισμένα στην ταυτόχρονη επεξεργασία πολλών εργασιών με πολλή Ε/Ε
  1. Εργασίες δέσμης (batch processing)
  2. Εργασίες συναλλαγών (Transaction processing)
  3. Χρονομερισμός (Timesharing)
- Παράδειγμα ΛΣ: OS/360 (αντικαθίστανται σταδιακά από UNIX παραλλαγές – Linux)

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- Χιλιάδες GB δεδομένα και εκατοντάδες δίσκους.
- Εργασίες δέσμης (batch processing):
  - Επαναλαμβανόμενες εργασίες χωρίς αλληλεπίδραση με χρήστες (π.χ. επεξεργασία στοιχείων τραπεζικών κινήσεων).
- Εργασίες συναλλαγών (Transaction processing):
  - Μεγάλος αριθμός από μικρές αιτήσεις χρηστών (π.χ. κρατήσεις θέσεων).
- Χρονομερισμός (Timesharing):
  - Πολλοί χρήστες ταυτόχρονα (π.χ. ερωτήματα σε βάση).

ΛΣ εξυπηρετητών
(Servers)

- Μεγάλοι υπολογιστές.
- Εξυπηρετούν ταυτόχρονα πολλούς χρήστες, συνήθως μέσω διαδικτύου.
  - Εξυπηρετητές ιστού (web servers)
  - Εξυπηρετητές ταχυδρομείου (mail servers)
  - Εξυπηρετητές αρχείων (file servers) ...
- Τυπικά ΛΣ: Solaris, FreeBSD, Linux, Windows Server, κτλ.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

ΛΣ πολυεπεξεργαστικών συστημάτων

- Σύνδεση πολλών CPU σε ένα σύστημα.
- Απαιτήσεις διαχείρισης πολλαπλών CPU από ένα μηχάνημα – προσπάθεια βέλτιστης εκμετάλλευσης της επεξεργαστικής ισχύος.
- Πλέον και οι απλοί υπολογιστές χρησιμοποιούν επεξεργαστές πολλών πυρήνων (multi-core processors)

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

ΛΣ προσωπικών υπολογιστών

- Εξυπηρετούν τις ανάγκες απλών χρηστών.
- Οι σύγχρονοι προσωπικοί υπολογιστές υποστηρίζουν πολυπρογραμματισμό, πολυνημάτωση, πολλαπλούς επεξεργαστές κτλ.
- Linux, Windows, FreeBSD, MAC OS ...

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

ΛΣ υπολογιστών χειρός
(Handheld computers)

- Ένα σύγχρονο PDA ή έξυπνο κινητό τηλέφωνο, είναι ένας μικρός Η/Υ.
- Υποστηρίζουν πλέον πολυπρογραμματισμό
- Μικρότερη οθόνη, πληκτρολόγιο (πλέον και αφής)
- CPU 32 bit
- Σκληρούς δίσκους μέχρι δεκάδων GB
- Symbian OS, Android, Iphone OS, Palm OS, Windows Mobile, ...

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- …με δυνατότητες αντίστοιχες προσωπικών Η/Υ της αμέσως προηγούμενης γενιάς.

ΛΣ Ενσωματωμένων συστημάτων
(Embedded systems)

- ΛΣ που ελέγχουν συσκευές ειδικού σκοπού (τηλεόραση, οικιακές συσκευές, αυτοκίνητα κτλ).
- Εκτελούν μόνο συγκεκριμένο λογισμικό του κατασκευαστή.
  - Δεν είναι υπολογιστές γενικού σκοπού.
  - Δεν απειλείται από μη έμπιστο λογισμικό.
- QNX, Vx Works αλλά και Linux.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

ΛΣ κόμβων αισθητήρων
(Sensor nodes)

- Κόμβοι με πολύ χαμηλές δυνατότητες σε CPU,μνήμη, ενέργεια κτλ).
- Ασύρματη επικοινωνία.
- Απαιτείται μεγάλη ανθεκτικότητα σε απειλές.
- Τα ΛΣ αισθητήρων είναι πολύ μικρά:
  - Καθοδηγούμενα από γεγονότα (event-driven)
  - Οι χρήστες δεν εκκινούν προγράμματα της επιλογής τους.
- Παράδειγμα ΛΣ: Tiny OS.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

ΛΣ πραγματικού χρόνου
(real-time systems)

- Ο χρόνος είναι η πιο σημαντική παράμετρος (βιομηχανικά συστήματα, αλυσίδες παραγωγής κτλ)
- Hard-time: Απόλυτη τήρηση του χρονοδιαγράμματος (π.χ. αεροσκάφη).
- Soft-time: Περιστασιακή μη τήρηση είναι αποδεκτή (π.χ. εικόνα, ήχος).
- Παράδειγμα ΛΣ: e-Cos.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

ΛΣ έξυπνων καρτών
(smart cards)

- Κάρτες με περιορισμένη CPU και μνήμη.
- Εκτελούν συγκεκριμένες λειτουργίες (π.χ. πληρωμή).
- Συνήθως εκτελεί εφαρμογές Java (java applets) πάνω σε διερμηνευτή (Java Virtual Machine).
- Το ΛΣ είναι αρκετά «πρωτόγονο» για εργασίες όπως πολυπρογραμματισμός ή διαχείριση πόρων.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

1.5 Βασικές έννοιες Λ.Σ.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Βασικές έννοιες των ΛΣ

- Διεργασίες (Processes)
- Χώροι διευθύνσεων(Address spaces)
- Αρχεία (Files)
- Είσοδος/Έξοδος (Input/Output)
- Προστασία (Protection)
- Κέλυφος (shell)

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Διεργασίες

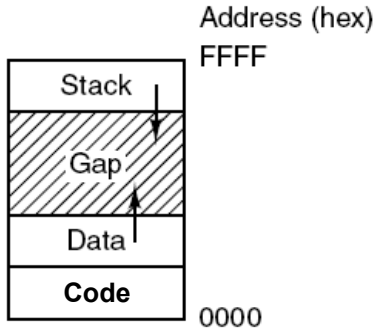
- Διεργασία = ένα πρόγραμμα που εκτελείται.
  Αποτελείται από:
  - Το **χώρο διευθύνσεων (address space)** της διεργασίας στη μνήμη που περιλαμβάνει τον **κώδικα** του προγράμματος που εκτελείται, τα **δεδομένα** του προγράμματος και τη **στοίβα**.
  - **Καταχωρητές**: όπως ο **μετρητής προγράμματος,** και ο **δείκτης στοίβας.**
  - Ανοικτά αρχεία, προειδοποιήσεις, λίστα σχετιζόμενων διεργασιών κτλ.
- Το ΛΣ διαχειρίζεται πολλές διεργασίες, διατηρώντας ένα **πίνακα διεργασιών (process table).**

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- Στο **χώρο διευθύνσεων** η διεργασία μπορεί να διαβάσει και να γράψει.

- Ο **πίνακας διεργασιών (Process table)** μπορεί να είναι ένας πίνακας ή μία συνδεδεμένη λίστα.

Η διεργασία στη μνήμη

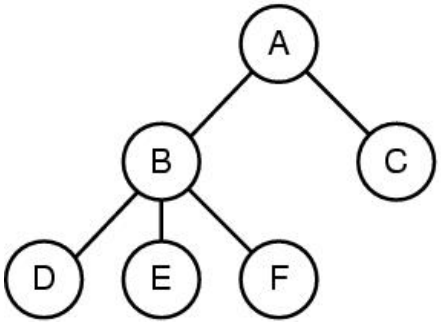


Εικόνα 1-20. Οι διεργασίες έχουν τρία τμήματα μνήμης: πρόγραμμα (code), δεδομένα (data), και στοίβα (stack).

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- Στο UNIX η μνήμη που αναλογεί σε κάθε διεργασία, διαιρείται σε 3 μέρη όπως φαίνεται στην εικόνα.
- Η στοίβα επεκτείνεται στο χάσμα αυτόματα (προς τα κάτω).
- Τα δεδομένα επεκτείνονται στο χάσμα ρητά μόνο μετά από την κλήση συστήματος **brk.**
- Όμως **η brk δεν ορίζεται στο POSIX.**
- Οι προγραμματιστές ενθαρρύνονται να χρησιμοποιήσουν την **malloc** για δυναμική κατανομή της μνήμης.

Δένδρο διεργασιών



Εικόνα 1-13. Η διεργασία A δημιουργεί δύο διεργασίες-παιδιά (ή θυγατρικές), την B και την C. Η διεργασία B δημιουργεί με τη σειρά της άλλες τρεις, τις D, E, και F.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- Οι διεργασίες μπορούν να δημιουργηθούν:
  1. Είτε από το χρήστη μέσω μίας εντολής π.χ. μέσω του **διερμηνευτή διαταγών (command interpreter)** ή μέσω του **κελύφους (shell)**.
  2. Είτε από μία άλλη διεργασία, το οποίο οδηγεί σε μία δενδροειδή δομή, δημιουργώντας **θυγατρικές διεργασίες**.

- Οι διεργασίες ενός δένδρου πρέπει να επικοινωνούν μεταξύ τους – **Δια-διεργασιακή Επικοινωνία** (Inter-process Communication).
  - Το ΛΣ μπορεί να στέλνει **σήματα συναγερμού (alarm signals)** σε μία διεργασία, ώστε να αναστείλει την εκτέλεσή της.

Χώροι διευθύνσεων (Μνήμη)

- Κάθε πρόγραμμα που εκτελείται, πρέπει να βρίσκεται στην κύρια μνήμη.
  - Το ΛΣ υποστηρίζει το χώρο διευθύνσεων των διεργασιών.
- Όταν βρίσκονται πολλά προγράμματα στη μνήμη ταυτόχρονα, απαιτείται προστασία.
- Τι συμβαίνει όταν ένα πρόγραμμα απαιτεί περισσότερο χώρο διευθύνσεων από ότι διαθέτει το σύστημα;
  - Χρήση **Εικονικής μνήμης (Virtual memory).**

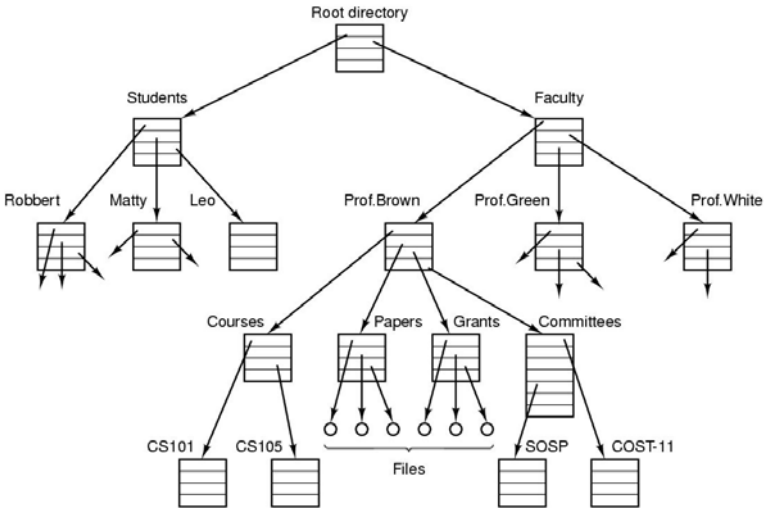
Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Εικονική μνήμη:

•το ΛΣ δημιουργεί χώρο στο δίσκο τον οποίο τον χρησιμοποιεί σαν να ήταν χώρος στη μνήμη.

•Ανάλογα με τις ανάγκες εκτέλεσης προγραμμάτων, μετακινεί τμήματα από την κύρια στην εικονική μνήμη και αντίστροφα.

Αρχεία

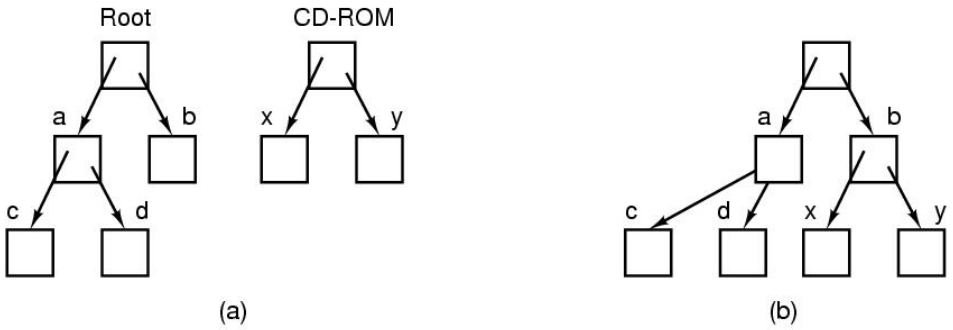


Εικόνα1-14. Ένα σύστημα αρχείων ενός τμήματος πανεπιστημίου.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

- Αφαίρεση για την δημιουργία, ανάγνωση, εγγραφή δεδομένων.
- Τα αρχεία δομούνται από το ΛΣ σε δένδρα τα οποία περιλαμβάνουν **καταλόγους (directories)** σε πολλά επίπεδα.
- Για την πρόσβαση σε ένα αρχείο χρησιμοποιούνται ειδικές **κλήσεις συστήματος**.
    - Για να ελεγχθούν τα δικαιώματα και η πρόσβαση σε ένα αρχείο, το ΛΣ αντιστοιχίζει έναν μικρό ακέραιο σε κάθε αρχείο που λέγεται **περιγραφέας αρχείου (file descriptor).**

Αρχεία
(ανάρτηση - mount)

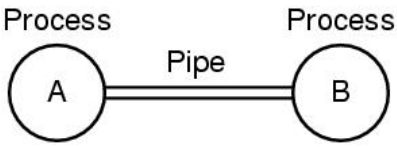


Εικόνα1-15. (a) Πριν την φόρτωση (mounting), τα αρχεία στο CD-ROM δεν είναι προσβάσιμα. (b) Μετά τη φόρτωση (mounting), αποτελούν κλαδί της ιεραρχίας των αρχείων.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- Στο UNIX όλα τα αρχεία έχουν μία κοινή ρίζα (σε αντίθεση με τα windows).
- Αυτό επιτυγχάνεται με την **ανάρτηση (mount)** μίας μονάδας δίσκου (π.χ. CD) στον βασικό κατάλογο.
- Το UNIX έχει επίσης τα **ειδικά αρχεία (special files)** τα οποία επιτρέπουν στις συσκευές E/E να φαίνονται σαν αρχεία.
    - Τα ειδικά αρχεία μπλοκ μοντελοποιούν συσκευές που αποτελούνται από μπλοκ (δίσκοι).
    - Τα ειδικά αρχεία χαρακτήρων μοντελοποιούν συσκευές που χειρίζονται ακολουθίες χαρακτήρων (π.χ. εκτυπωτές).

Αρχεία
(Αγωγοί ή σωληνώσεις - pipes)



Εικόνα 1-16. Δύο διεργασίες που συνδέονται με μία σωλήνωση (pipe).

Παράδειγμα:

cat file1 file2 file3 | sort >/del/lp

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- Μία σωλήνωση ή αγωγός (pipe) είναι ένα ψευδο-αρχείο το οποίο επιτρέπει σε διεργασίες να επικοινωνήσουν.
- Δημιουργείται προκαταβολικά. Η διεργασία Α γράφει στον αγωγό σαν να ήταν αρχείο και η διεργασία Β διαβάζει από τον αγωγό.
- Στο παράδειγμα, συνενώνονται τα αρχεία file1, file2, file3 με την εντολή **cat** (1η διεργασία). Τα αποτέλεσμα το στέλνει η σωλήνωση στη 2η διεργασία που εκκινεί η εντολή **sort** ώστε να ταξινομήσει τα δεδομένα και τα ανακατευθύνει στον εκτυπωτή.

1.6 Κλήσεις συστήματος

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Κλήσεις συστήματος (System Calls)

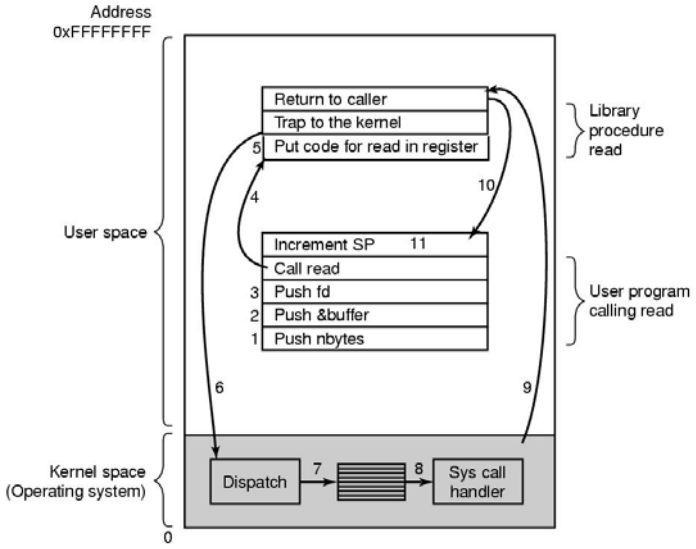
- Θυμηθείτε ότι το ΛΣ εκτελείται σε κατάσταση πυρήνα (kernel mode) και όλο το υπόλοιπο λογισμικό σε κατάσταση χρήστη (user mode).
- Για να χρησιμοποιήσει μία διεργασία μία υπηρεσία του ΛΣ (π.χ. ανάγνωση από αρχείο) θα εκτελέσει μία **κλήση συστήματος (system call)**.
- Κάθε κλήση συστήματος υλοποιείται μέσω μίας ρουτίνας που βρίσκεται σε μία βιβλιοθήκη διαδικασιών (συνήθως σε C).
- Η κλήση συστήματος προκαλεί μία **παγίδευση (trap)** και μεταβαίνει σε κατάσταση πυρήνα.
- Η εντολή TRAP προκαλεί την ανάμιξη του ΛΣ.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

- Η εντολή TRAP προκαλεί αλλαγή από κατάσταση χρήστη σε κατάσταση πυρήνα και ξεκινά την ανάμιξη του ΛΣ. Όταν η εργασία ολοκληρωθεί, ο έλεγχος επιστρέφει στο πρόγραμμα χρήστη, στην εντολή που ακολουθεί την κλήση συστήματος.
  - Π.χ. fork(), exec(), malloc(), read(), printf()

- Κάθε κλήση συστήματος υλοποιείται μέσω μίας **ρουτίνας** που βρίσκεται σε μία **βιβλιοθήκη** που συνδέεται (linked) με τον κώδικα της user process. Όλες οι ρουτίνες αυτής της βιβλιοθήκης εκτελούν μία ειδική εντολή, που ονομάζεται TRAP. Η εντολή TRAP είναι αυτή η οποία αλλάζει το σύστημα από user mode σε kernel mode, (αλλάζοντας ένα bit στον καταχωρητή PSW της CPU).

Κλήσεις συστήματος
(Παράδειγμα)



Εικόνα 1-17. Τα 11 βήματα της κλήσης συστήματος **read(fd, buffer, nbytes)**

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Η διαδικασία (ρουτίνα) βιβλιοθήκης *read* (σε C) εκκινεί την κλήση συστήματος **`read`** (σε συμβολική γλώσσα).

1-3: Ο μεταγλωττιστής της C ή C++ τοποθετεί τις μεταβλητές στη στοίβα (by value, με & by reference).

4: Γίνεται η κλήση της ρουτίνας βιβλιοθήκης *read* ().

5: Η ρουτίνα βιβλιοθήκης *read* τοποθετεί τον αριθμό της κλήσης συστήματος, σε ένα καταχωρητή που γνωρίζει το ΛΣ.

6: Εκτελείται η εντολή TRAP και γίνεται μετάβαση σε κατάσταση πυρήνα.

7: Ο κώδικας του πυρήνα που εκτελείται μετά την TRAP εξετάζει τον αριθμό της κλήσης συστήματος και μεταβιβάζει τον έλεγχο στον κατάλληλο χειριστή κλήσεων συστήματος.

8: Εκτέλεση του χειριστή συστήματος.

9: Τέλος της εργασίας, επιστροφή στη ρουτίνα (διαδικασία) βιβλιοθήκης.

10: Η ρουτίνα βιβλιοθήκης επιστρέφει στο πρόγραμμα χρήστη.

11: Το πρόγραμμα χρήστη αυξάνει το δείκτη στοίβας (Stack Pointer) ακριβώς τόσο ώστε να καθαρίσει η στοίβα από τις μεταβλητές της *read*.

Κλήσεις συστήματος για τη Διαχείριση Διεργασιών

Process management

Call	Description
pid = fork( )	Create a child process identical to the parent
pid = waitpid(pid, &statloc, options)	Wait for a child to terminate
s = execve(name, argv, environp)	Replace a process' core image
exit(status)	Terminate process execution and return status

Εικόνα1-18. Μερικές από τις κυριότερες κλήσεις συστήματος του POSIX.

pid: process ID (η ταυτότητα της διεργασίας)
s: κωδικός σφάλματος (-1 για λάθος)
fd: file descriptor (περιγραφέας αρχείου)

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

1) Δημιουργία μίας θυγατρικής διεργασίας, όμοιας με τη γονική
   - Ο μοναδικός τρόπος δημιουργίας μίας διεργασίας στο UNIX
   - Η fork επιστρέφει την τιμή **0 στη θυγατρική διεργασία**, και την τιμή pid στη γονική.
2) Αναμονή μέχρι τον τερματισμό της θυγατρικής διεργασίας
   - Στη διεύθυνση που δείχνει η 2η παράμετρος statloc τοποθετείται ο κωδικός εξόδου (exit status ή κωδικός σφάλματος).
3) Αντικαθιστά την εικόνα πυρήνα (δηλαδή το χώρο διευθύνσεων της διεργασίας στη μνήμη)
   - Μετά τη fork() έχει δημιουργηθεί η θυγατρική διεργασία. Η execve( ) αντικαθιστά την εικόνα πυρήνα (χώρο διευθύνσεων) της θυγατρικής διεργασίας, με το αρχείο που αναφέρεται στην name.
4) Τερματίζει μία διεργασία και επιστρέφει την κατάσταση εξόδου.

Ένα απλό κέλυφος

```
#define TRUE 1

while (TRUE) {                                  /* repeat forever */
      type_prompt( );                           /* display prompt on the screen */
      read_command(command, parameters);        /* read input from terminal */

      if (fork( ) != 0) {                       /* fork off child process */
           /* Parent code. */
           waitpid(–1, &status, 0);             /* wait for child to exit */
      } else {
           /* Child code. */
           execve(command, parameters, 0);      /* execute command */
      }
}
```

Εικόνα 1-19. Παράδειγμα ενός απλού κελύφους (command shell).

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- Περιμένει να λάβει εντολή.
- Μόλις τη λάβει, διαβάζει τις παραμέτρους και:
- Εάν fork() =! 0 είμαστε στη γονική διεργασία (θυμηθείτε ότι η θυγατρική παίρνει pid=0)
    - Περιμένει μέχρι να τελειώσει η θυγατρική.

- Η θυγατρική καλεί την κλήση συστήματος execve με ορίσματα την εντολή και τις παραμέτρους που δόθηκαν. Αλλάζει έτσι το χώρο διευθύνσεών της στη μνήμη, εκτελώντας τον κώδικα της εντολής.

(Παράδειγμα διαταγής)

cp file1 file2

- Το κύριο πρόγραμμα που υλοποιεί την εντολή cp περιέχει τη δήλωση: **main(argc, argv, envp)**
  - argc: το πλήθος των ορισμάτων (3)
  - argv: δείκτης σε ένα πίνακα
    - argv[0] → cp, argv[1] → file1, argv[2] → file2
  - envp: enviroment variables (π.χ. όνομα home directory)

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Κλήσεις συστήματος για τη Διαχείριση Αρχείων

File management

Call	Description
fd = open(file, how, ...)	Open a file for reading, writing, or both
s = close(fd)	Close an open file
n = read(fd, buffer, nbytes)	Read data from a file into a buffer
n = write(fd, buffer, nbytes)	Write data from a buffer into a file
position = lseek(fd, offset, whence)	Move the file pointer
s = stat(name, &buf)	Get a file's status information

Εικόνα 1-18. Μερικές από τις κυριότερες κλήσεις συστήματος του POSIX.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Στο open:Το how μπορεί να είναι *O_RDONLY, O_WONLY, O_RDWR, O_CREAT*

Στο lseek: fd: file descriptor, offset: μία θέση στο αρχείο, whence: η θέση του Offset υπολογίζεται από την αρχή, το τέλος ή την τρέχουσα θέση του αρχείου;

Κλήσεις συστήματος για τη Διαχείριση Καταλόγων

Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

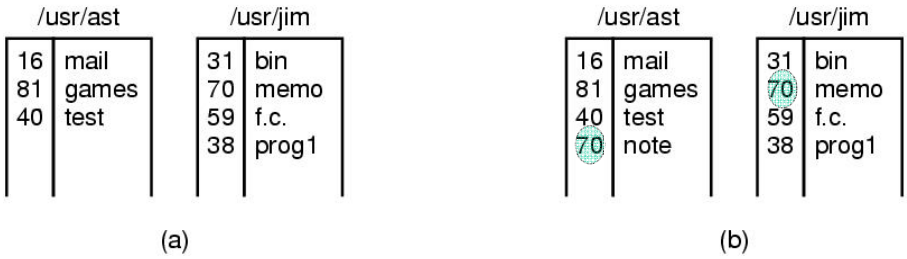
Εικόνα 1-18. Μερικές από τις κυριότερες κλήσεις συστήματος του POSIX.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

Η link: επιτρέπει την εμφάνιση ενός αρχείο σε περισσότερες θέσεις, με διαφορετικά ονόματα (όμως διατηρεί το ί-node number).

Σύνδεσμοι αρχείων (linking)

Παράδειγμα κλήσης link:
link ("usr/jim/memo", "/usr/ast/note");

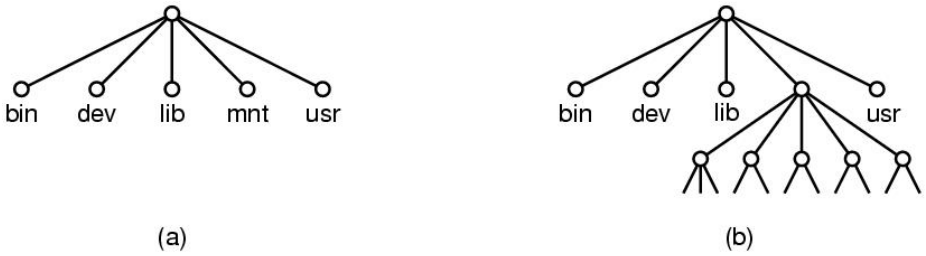


Εικόνα 1-21. (a) Δύο κατάλογοι πριν τη σύνδεση του */usr/jim/memo* στον κατάλογο ast.
(b) Οι ίδιοι κατάλογοι μετά τη σύνδεση.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Φόρτωση αρχείων (Mounting)

Παράδειγμα κλήσης mount:
mount("/dev/hda", "/mnt", 0);



Εικόνα 1-22. (a) Το σύστημα αρχείων πριν τη φόρτωση. (b) Το σύστημα αρχείων μετά τη φόρτωση.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Διάφορες κλήσεις συστήματος

Call	Description
s = chdir(dirname)	Change the working directory
s = chmod(name, mode)	Change a file's protection bits
s = kill(pid, signal)	Send a signal to a process
seconds = time(&seconds)	Get the elapsed time since Jan. 1, 1970

Εικόνα 1-18. Μερικές από τις κυριότερες κλήσεις συστήματος του POSIX.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Το API Win32 των Windows (Application Programming Interface)

UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time

Εικόνα 1-23. Οι Win32 API κλήσεις σε, κατά προσέγγιση, αντιστοίχηση με τις κλήσεις συστήματος του UNIX.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

UNIX: όλα μέσω συγκεκριμένων κλήσεων συστήματος.

Windows: event-driven

•Το κύριο πρόγραμμα περιμένει κάποιο συμβάν και καλεί την αντίστοιχη διαδικασία να το χειριστεί.

•Υπάρχουν και εδώ κλήσεις συστήματος, αλλά σε αντίθεση με το Unix, η αντιστοιχία κλήσεων συστήματος και διαδικασιών βιβλιοθήκης, δεν είναι 1 προς 1.

•Η Microsoft ορίζει ένα σύνολο από διαδικασίες με το όνομα Win32 API (Application Programming Interface) το οποίο χρησιμοποιούν οι προγραμματιστές για να έχουν πρόσβαση στις υπηρεσίες του ΛΣ.

•Εφόσον διαχωρίζεται η πραγματική υλοποίηση από την διασύνδεση, η microsoft μπορεί να αλλάζει την υλοποίηση της κλήσης από έκδοση σε έκδοση, χωρίς να αλλάζει τη διασύνδεσή της.

•Αριθμός κλήσεων Windows API: χιλιάδες.

- Ενώ πολλές πραγματοποιούν κλήσεις συστήματος, άλλες εκτελούνται εξολοκλήρου στο χώρο του χρήστη!
- Είναι αδύνατο να γνωρίζει κανείς αν μία κλήση είναι συστήματος (εκτελείται στον πυρήνα) ή απλή κλήση χρήστη.

1.7 Η δομή των λειτουργικών συστημάτων

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Δομές Λειτουργικών Συστημάτων

Μονολιθικά συστήματα – βασική δομή:

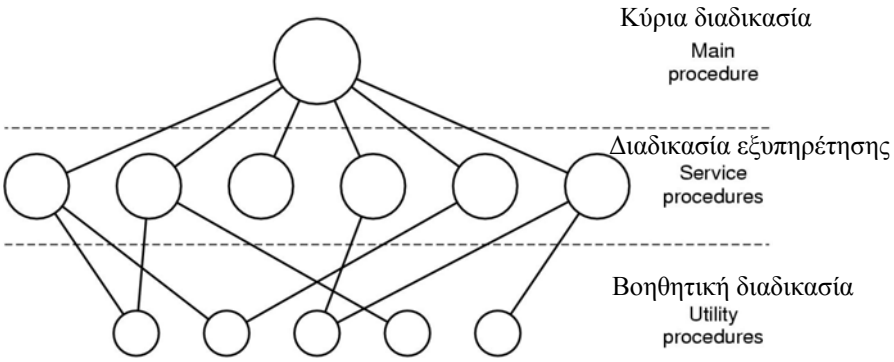
- Η πιο διαδεδομένη οργάνωση.
- Ένα κύριο πρόγραμμα το οποίο καλεί την αιτούμενη διαδικασία εξυπηρέτησης.
- Ένα σύνολο διαδικασιών υπηρεσιών (service procedures) οι οποίες εκτελούν τις κλήσεις συστήματος.
- Ένα σύνολο βοηθητικών διαδικασιών (utility procedures) οι οποίες βοηθούν τις διαδικασίες υπηρεσιών.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

•Το ΛΣ είναι γραμμένο ως μία συλλογή από διασυνδεδεμένων διαδικασιών, σε ένα μεγάλο, εκτελέσιμο πρόγραμμα.

- Αρχικά μεταγλωττίζονται όλες οι διαδικασίες και μετά συνενώνονται (linked).
- Κάθε διαδικασία, είναι ορατή από όλες τις άλλες.

Μονολιθικά συστήματα



Εικόνα 1-24. Ένα απλό μοντέλο της δομής ενός μονολιθικού συστήματος

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

•Για κάθε κλήση συστήματος, υπάρχει μία κύρια διαδικασία εξυπηρέτησης, η οποία είναι υπεύθυνη για αυτή και την εκτελεί.

•Οι βοηθητικές διαδικασίες εκτελούν εργασίες χρήσιμες για περισσότερες από μία διαδικασίες εξυπηρέτησης, π.χ. μεταφορά δεδομένων από ένα πρόγραμμα χρήστη.

Πολύ-επίπεδα συστήματα

Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

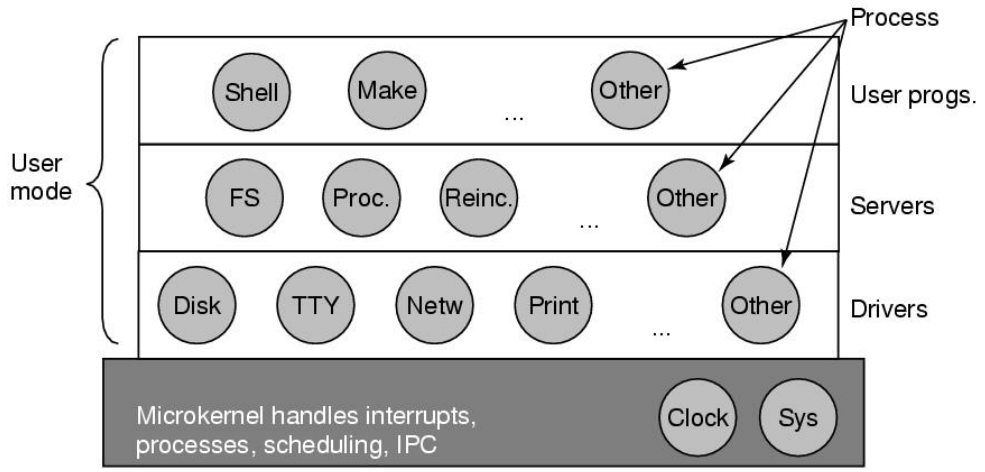
Εικόνα 1-25. Η δομή του ΛΣ «THE».

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

- Γενίκευση της προηγούμενης προσέγγισης είναι η δόμηση του ΛΣ σε πολλά ιεραρχικά επίπεδα (1ο το ΛΣ THE)
- Στην προσέγγιση αυτή, οι σχεδιαστές μπορούν να καθορίσουν τη θέση του ορίου μεταξύ πυρήνα και χρήστη.

1. Επίπεδο 0: διαμοίραση επεξεργαστή στις διεργασίες – πολυπρογραμματισμός.
2. Επίπεδο 1: Διαχείριση μνήμης (και δίσκου)
3. Επίπεδο 2: επικοινωνία μεταξύ διεργασίας και κονσόλας (κάθε διεργασία είχε τη δική της κονσόλα χειρισμού)
4. Επίπεδο 3: διαχείριση συσκευών Ε/Ε.
5. Επίπεδο 4: Προγράμματα χρήστη.
6. Επίπεδο 5: Χειριστής συστήματος.

Μικροπυρήνες (Microkernels)



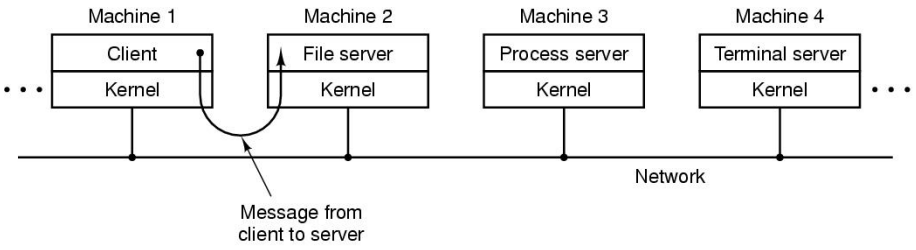
Εικόνα 1-26. Η δομή του ΛΣ «MINIX 3».

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

- Βασική λογική: τοποθέτηση όσο το δυνατό λιγότερων στοιχείων στον πυρήνα (επειδή τα σφάλματα στον πυρήνα μπορούν να προκαλέσουν κατάρρευση του συστήματος).
- Στόχος: υψηλή αξιοπιστία.
- Διαίρεση του ΛΣ σε μικρές, καλά ορισμένες υπομονάδες, από τις οποίες μόνο ο μικροπυρήνας εκτελείται σε κατάσταση πυρήνα.
- Οι υπόλοιπες εκτελούνται σε σχετικά λιγότερο ισχυρές διεργασίες χρήστη.

- Η επικοινωνία των διεργασιών χρήστη με συσκευές γίνεται μέσα από τους οδηγούς συσκευής.
  - Ο οδηγός συσκευής καθορίζει ποιες τιμές θα γραφτούν σε ποιες θύρες Ε/Ε και καλεί τον πυρήνα.
  - Σε αντίθεση με τα μονολιθικά, εάν ένας οδηγός συσκευής με σφάλματα, δεν θα μπορέσει να γράψει σε θέση που δεν επιτρέπεται.

Μοντέλο πελάτη-εξυπηρετητή (Client-Server)

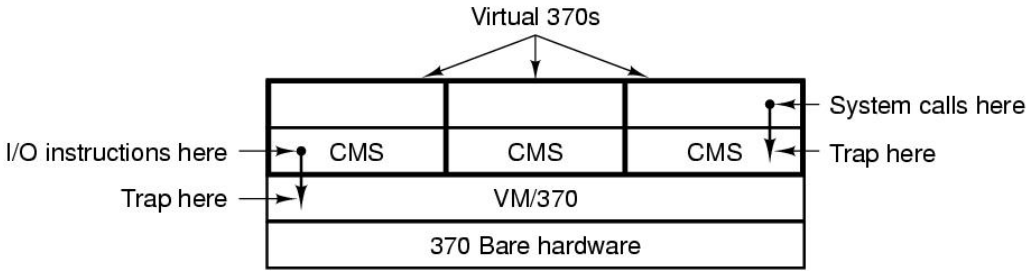


Εικόνα 1-27. Το μοντέλο πελάτη-εξυπηρετητή σε ένα δίκτυο.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

- Η παρουσία πυρήνα δεν είναι απαραίτητη:
- Διεργασίες πελάτη και διεργασίες εξυπηρετητή.
  - Η επικοινωνία μεταξύ διεργασιών πελάτη και διεργασιών εξυπηρετητή γίνεται με μηνύματα.
  - Εφοσον οι πελάτες και οι εξυπηρετητές είναι απομακρυσμένοι και επικοινωνούν μέσω μηνυμάτων, οι πελάτες δεν χρειάζεται να γνωρίζουν τις λεπτομέρειες στον εξυπηρετητή.

Εικονικές μηχανές (1)



Εικόνα 1-28. Η δομή του VM/370 με CMS.

Μετάφραση βασισμένη στις διαφάνειες της Αγγλικής έκδοσης
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

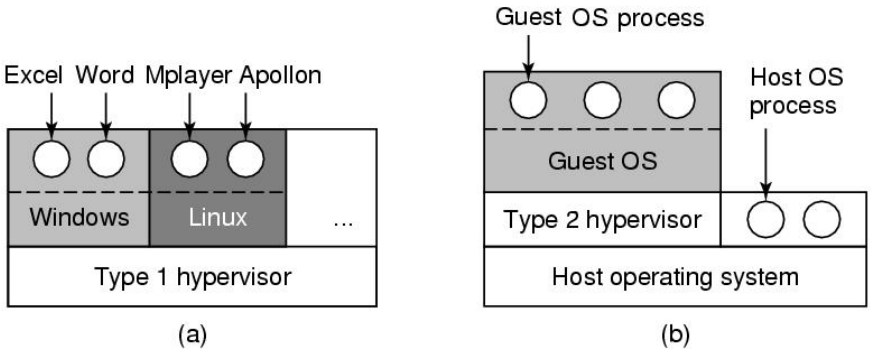
Σύστημα VM/370:

•**Διαχωρισμός πολυπρογραμματισμού και εκτεταμένης (εικονικής) μηχανής**.

•Βάση του συστήματος το **virtual machine monitor**, το οποίο εκτελείται απευθείας στο υλικό και υλοποιεί τον πολυπρογραμματισμό.

- Μπορεί να παρέχει πολλές εικονικές μηχανές ταυτόχρονα.
- Κάθε εικονική μηχανή είναι ακριβές αντίγραφο του ίδιου του υλικού
- Γίνονται δύο παγιδεύσεις: μία στην εικονική μηχανή (για την κλήση συστήματος) και μία στο VM monitor (για Ε/Ε στο υλικό).

Εικονικές μηχανές (2)



Εικόνα 1-29. (a) Υπερεπόπτης τύπου-1 (type 1 hypervisor). (b) Υπερεπόπτης τύπου-2 (type 2 hypervisor).

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

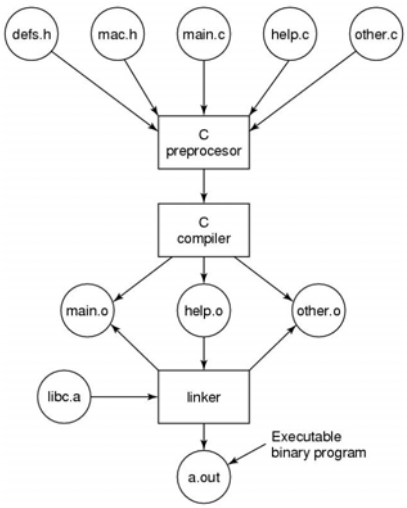
1.8 Βασικές έννοιες της γλώσσας C

Ο κόσμος της γλώσσας C

- Η γλώσσα προγραμματισμού C
- Αρχεία επικεφαλίδας (Header files)
- Μεγάλα έργα προγραμματισμού
- Το μοντέλο του χρόνου εκτέλεσης

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Το μοντέλο του χρόνου εκτέλεσης



Εικόνα 1-30. Η διαδικασία της μεταγλώττισης στη C και τα αρχεία επικεφαλίδας για τη δημιουργία ενός εκτελέσιμου αρχείου.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Μονάδες μέτρησης

Exp.	Explicit	Prefix	Exp.	Explicit	Prefix
10^{-3}	0.001	milli	10^{3}	1,000	Kilo
10^{-6}	0.000001	micro	10^{6}	1,000,000	Mega
10^{-9}	0.000000001	nano	10^{9}	1,000,000,000	Giga
10^{-12}	0.000000000001	pico	10^{12}	1,000,000,000,000	Tera
10^{-15}	0.000000000000001	femto	10^{15}	1,000,000,000,000,000	Peta
10^{-18}	0.000000000000000001	atto	10^{18}	1,000,000,000,000,000,000	Exa
10^{-21}	0.000000000000000000001	zepto	10^{21}	1,000,000,000,000,000,000,000	Zetta
10^{-24}	0.000000000000000000000001	yocto	10^{24}	1,000,000,000,000,000,000,000,000	Yotta

Περιγραφή των μονάδων μέτρησης