

# Εισαγωγή στα Λειτουργικά Συστήματα I

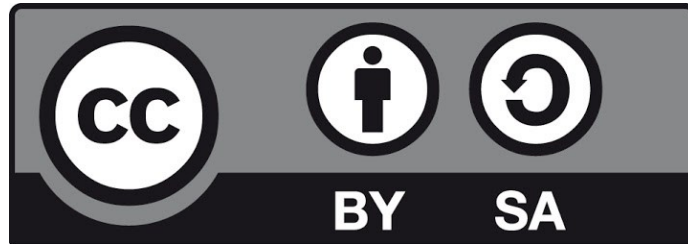
---

Εισηγητής: Χρήστος Δαλαμάγκας

[cdalamagkas@gmail.com](mailto:cdalamagkas@gmail.com)

# Άδεια χρήσης

Το παρόν εκπαιδευτικό υλικό υπόκειται στη διεθνή άδεια χρήσης Creative Commons Attribution-ShareAlike 4.0 (<https://creativecommons.org/licenses/by-sa/4.0/>).



# Βιβλιογραφικές πηγές

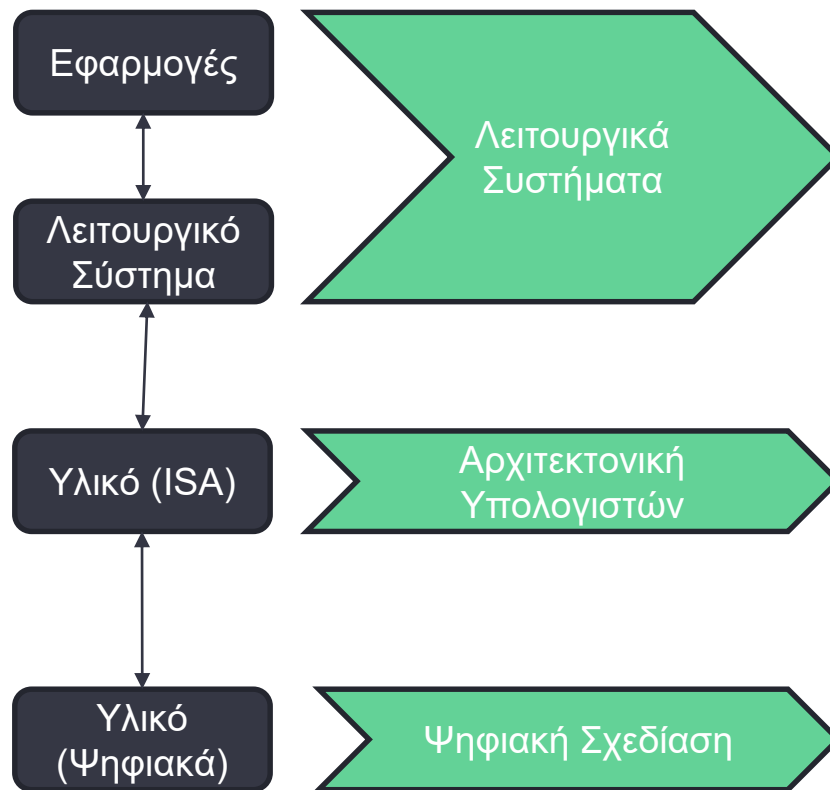
- Σημειώσεις κ. Χρήστου Παναγιωτόπουλου
- Σημειώσεις δρ.α Μηνά Δασυγένη, Πανεπιστήμιο Δυτικής Μακεδονίας  
<https://eclass.uowm.gr/courses/ICTE264/>
- Λοιπές πηγές

# Ύλη μαθήματος

- Θεωρητικό υπόβαθρο λειτουργικών συστημάτων
  - Ορισμός, Κατηγοριοποίηση
  - Ο πυρήνας του ΛΣ
- Διεργασίες και διαχείριση πόρων της ΚΜΕ
- Διαχείριση Μνήμης
- Αρχεία και συστήματα αρχείων
- Λειτουργικά συστήματα UNIX/LINUX
  - Χαρακτηριστικά και πλεονεκτήματα
  - Βασικές εντολές και παραμετροποίηση, βοηθητικά προγράμματα (backup, data recovery)
  - Κατάλογοι και δικαιώματα
- Λειτουργικά συστήματα Windows
  - Χαρακτηριστικά (registry)
  - Βασικές εντολές και παραμετροποίηση (backup, data recovery)
- Virtualization και κατηγοριοποίηση
  - Τύπος 2: VirtualBox, Vmware Workstation
  - Τύπος 1: Vmware ESXi, XenServer
  - Εικονικοποίηση επιπέδου λογισμικού (containerization)

# Ορισμός των ΛΣ

- Λογισμικό/διεπαφή μεταξύ του σετ εντολών assembly και των εφαρμογών
- Προσφέρεται ως **εικονική μηχανή** για διαχείριση και ανάθεση πόρων
- **Μηχανισμός αφαίρεσης** του υλικού και της ISA
- Εκτελείται σε κατάσταση πυρήνα



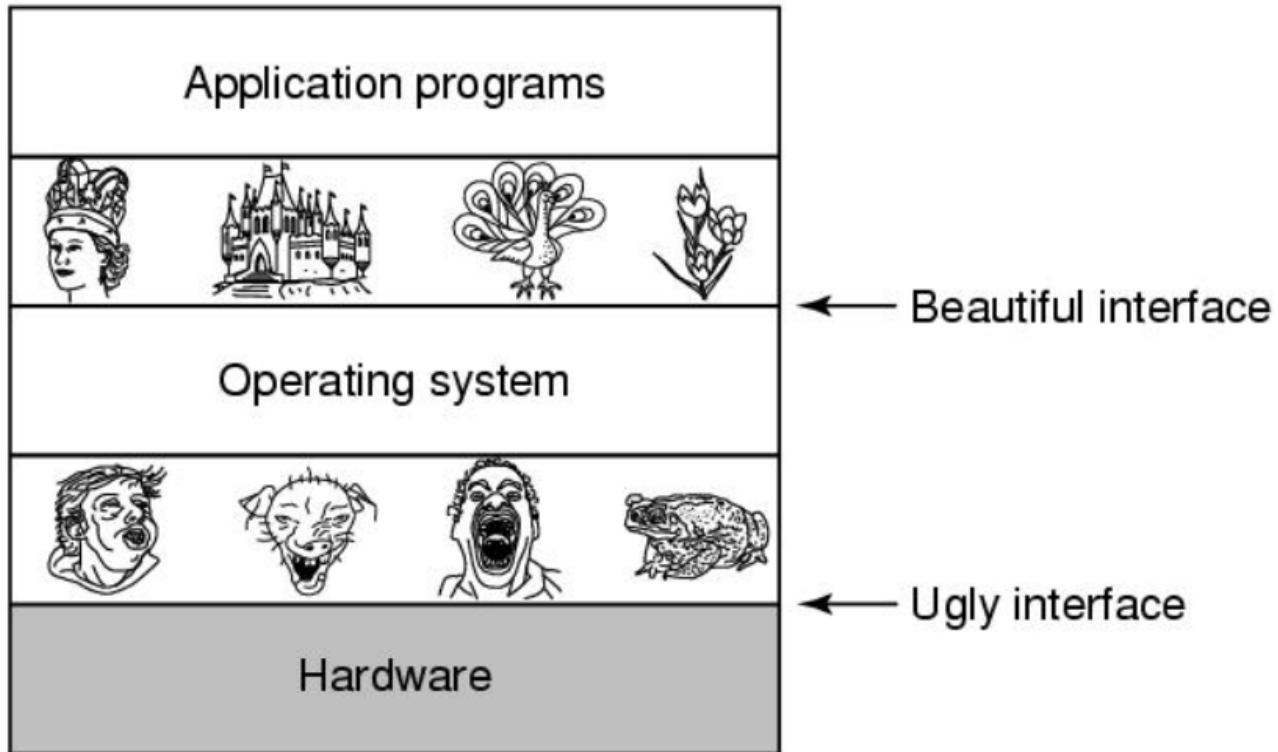
# Το ΛΣ ως εικονική μηχανή

- Το ΛΣ δημιουργεί την **ψευδαίσθηση** σε μια εφαρμογή πως είναι η μόνη που εκτελείται στο υλικό.
- Παρέχει μηχανισμούς για απομόνωση των διεργασιών.
- Δυνητικά, μια εφαρμογή μπορεί να χρησιμοποιήσει όλους τους πόρους ενός συστήματος.
- Το ΛΣ μας επιτρέπει να εκτελούμε ταυτόχρονα περισσότερες διεργασίες, λόγω αυτής της ψευδαίσθησης.
- Επιτρέπει τη χωρική και χρονική πολυπλεξία των εφαρμογών

# Το ΛΣ ως μηχανισμός αφαίρεσης

- «Κρύβει» από τον προγραμματιστή τις λεπτομέρειες του υλικού.
- Ο προγραμματιστής προγραμματίζει σε υψηλότερο επίπεδο.
- Εφαρμόζονται αυτοματοποιημένες διαδικασίες για διαχείριση υλικού που δεν αφορούν τον προγραμματιστή.
- Παρέχει ένα σύνολο κλήσεων συστήματος (API) στον προγραμματιστή για επικοινωνία με το ίδιο το ΛΣ και το υλικό.
  - `fwrite` για εγγραφή αρχείων στον δίσκο
  - `malloc` για τη δέσμευση μνήμης RAM

Με μια εικόνα...





# Σχέση του ΛΣ με το υλικό

- Εξάρτηση μεταξύ ΛΣ και του Συνόλου Εντολών Αρχιτεκτονικής (Instructions Set Architecture – ISA)
- Οι δυνατότητες ενός ΛΣ εξαρτώνται από το ISA
- Νέες δυνατότητες στο υλικό και την ISA απαιτούν αντίστοιχη υποστήριξη από το ΛΣ

# Στόχοι των ΛΣ

- Παροχή εύχρηστης διεπαφής API στους προγραμματιστές για ανάπτυξη εφαρμογών.
- Αποδοτική διαχείριση των υπολογιστικών πόρων προς όφελος των εφαρμογών και του υπολογιστικού συστήματος.
- Προστασία των διεργασιών και των δεδομένων του συστήματος.

# Συστατικά των ΛΣ

---

# Ιστορία των ΛΣ

- Σειριακή επεξεργασία δέσμης (job batching)
  - Job: Ένα ή περισσότερα προγράμματα γραμμένα στην ίδια γλώσσα
  - Batching: Ομαδοποίηση παρόμοιων εργασιών και υποβολή τους στον Η/Υ προς εκτέλεση
  - Τα προγράμματα εκτελούνταν αυστηρά σειριακά
- Πολυπρογραμματισμός (multiprogramming)
  - Πολλά προγράμματα φορτώνονται στη μνήμη
  - Η ΚΜΕ εκτελεί ψευδοπαράλληλα όλα τα προγράμματα
  - Spooling: Επιτρέπονταν οι διαδικασίες I/O παράλληλα με την εκτέλεση των προγραμμάτων
  - Καταμερισμός χρόνου: Πόσο χρόνο θα πρέπει να διαθέσει η ΚΜΕ για κάθε διεργασία;

# Συστατικά ενός ΛΣ (1/2)

- Διεργασίες: Προγράμματα του χρήστη τα οποία βρίσκονται σε εκτέλεση στη RAM
- Πυρήνας: Το πρόγραμμα που εκτελεί τις βασικές λειτουργίες του ΛΣ
  - Διαχείριση μνήμης: Χωρίζει τη μνήμη σε τμήματα και τις αναθέτει κατάλληλα ώστε να προστατεύσει τις διεργασίες.
  - Χρονοπρογραμματισμός διεργασιών: Το ΛΣ αναθέτει επεξεργαστικούς πόρους στην κάθε διεργασία με δίκαιο και αποδοτικό τρόπο
  - Σύστημα αρχείων (VFS): Ορίζει πως τα δεδομένα αποθηκεύονται και ανακτώνται από τον δίσκο.
  - Δικτύωση: Οι λειτουργίες που επιτρέπουν την επικοινωνία μέσω TCP/IP
  - Οδηγοί: Υλικολογισμικό που παρέχει οδηγίες σε εσωτερικές ή εξωτερικές συσκευές/συστήματα

# Συστατικά ενός ΛΣ (1/2)

- Διακοπές: Σήματα που στέλνονται από διεργασίες ή υλικό προς την ΚΜΕ
- Καταστάσεις λειτουργίας
  - User space: Όλος ο κώδικας που εκτελείται «πάνω» από τον πυρήνα, με περιορισμένα δικαιώματα και απομόνωση.
  - Kernel mode: Ο κώδικας του πυρήνα που έχει το σύνολο του ελέγχου του ΥΣ
- Ασφάλεια:
  - Μηχανισμός δικαιωμάτων χρηστών
  - Διαβάθμιση δικαιωμάτων
  - Τείχος ασφαλείας και Λίστες Ελέγχου Πρόσβασης
- Διεπαφή χρήστη
  - Διεπαφή γραμμής εντολών - Κέλυφος
  - Γραφικό (παραθυρικό) περιβάλλον

# Παράλληλα συστήματα

- Κάνει την παράλληλη επεξεργασία πραγματικότητα
- Συστήματα με πολλές ΚΜΕ στο ίδιο υπολογιστικό σύστημα
- Ισχυρά συζευγμένα συστήματα (TCS): Οι επεξεργαστές μοιράζονται δίαυλο, ρολόι και μνήμη.
- Συμμετρική πολυεπεξεργασία (SMP): Όλες οι ΚΜΕ αντιμετωπίζονται το ίδιο
- Ασύμμετρη πολυεπεξεργασία (AMP): Διαφορετικοί ρόλοι για κάθε ΚΜΕ
- Προσφέρουν:
  - Αυξημένη απόδοση
  - Οικονομία κλίμακας
  - Αυξημένη αξιοπιστία

# Κατανεμημένα Συστήματα

- Κατανομή των εργασιών μεταξύ ΚΜΕ που βρίσκονται σε διαφορετικά υπολογιστικά συστήματα
- Χαλαρά συζευγμένα συστήματα (LCS): Κάθε ΚΜΕ έχει τη δική της τοπική μνήμη και επικοινωνεί με τις άλλες ΚΜΕ μέσω δικτυακών πρωτοκόλλων
- Προσφέρουν
  - Διαμοιρασμό πόρων
  - Επιτάχυνση υπολογισμών
  - Εξισορρόπηση φόρτου
  - Αξιοπιστία και διαθεσιμότητα
  - Κλιμακωσιμότητα

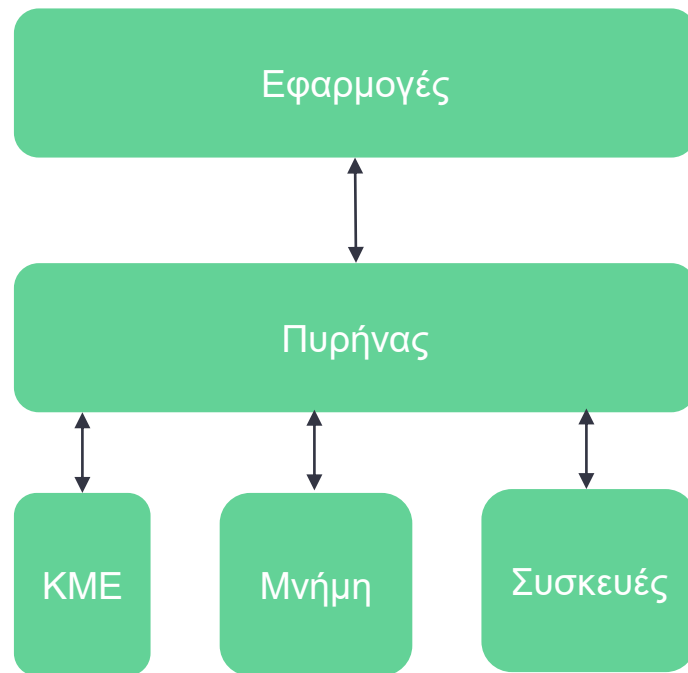


Πυρήνες των ΛΣ

---

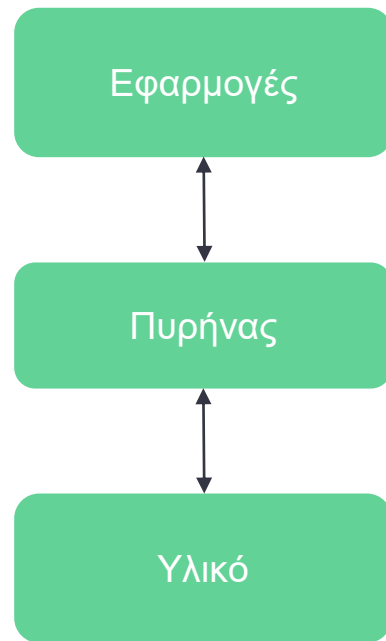
# Πυρήνας (kernel)

- Το κυρίως πρόγραμμα κάθε λειτουργικού συστήματος
- Συνδέει το υλικό με τις εφαρμογές
- Εκτελείται σε ειδικό χώρο της μνήμης RAM με δικαιώματα kernel
- Το μοναδικό πρόγραμμα με τόσα δικαιώματα
- Παρέχει το API για την υλοποίηση κλήσεων συστημάτων



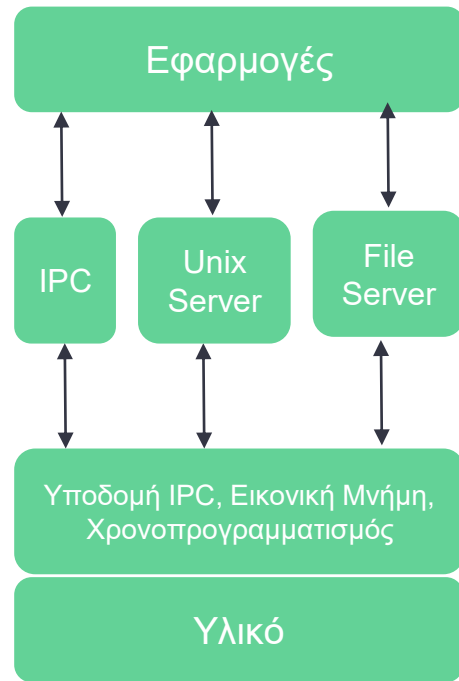
# Μονολιθικοί πυρήνες (monolithic)

- Το ΛΣ αποτελείται από ένα μεγάλο κομμάτι κώδικα
- Παρέχουν μια εκτεταμένη διεπαφή API για πρόσβαση στις λειτουργίες του συστήματος
- Υπάρχει μεγάλη εξάρτηση μεταξύ τμημάτων του μονολιθικού πυρήνα
- Παράδειγμα ΛΣ: VMS, Linux
- Γνωστοί υποστηρικτές: **Linus Torvalds**



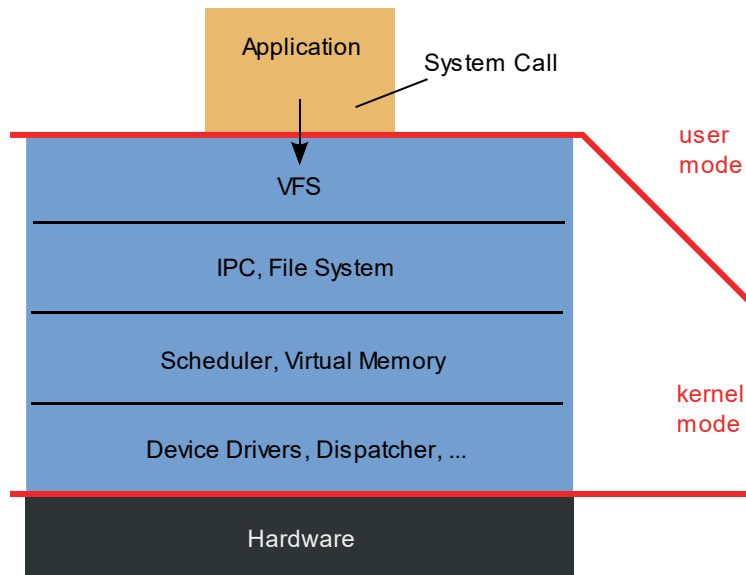
# Μικροπυρήνες (microkernel)

- Μικρότεροι από τους μονολιθικούς πυρήνες
- Οι ελάχιστες δυνατές λειτουργίες για την υλοποίηση ενός ΛΣ εκτελούνται σε κατάσταση πυρήνα: Βασική διαδικεργασική επικοινωνία, εικονική μνήμη και χρονοπρογραμματισμός.
- Οι υπόλοιπες λειτουργίες ανατίθενται σε διεργασίες/ εξυπηρετητές που εκτελούνται σε κατάσταση χρήστη
- Οι διεργασίες του χρήστη ονομάζονται πελάτες (clients)
- Ο μικροπυρήνας διαχειρίζεται την επικοινωνία μεταξύ πελατών και εξυπηρετητών
- Παράδειγμα ΛΣ: MINIX
- Γνωστοί υποστηρικτές: **Andrew S. Tanenbaum**

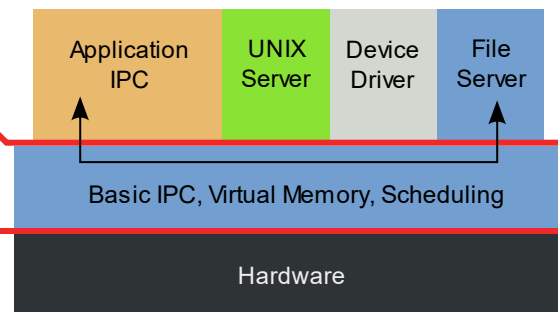


# Σύγκριση των δυο αρχιτεκτονικών

Monolithic Kernel  
based Operating System

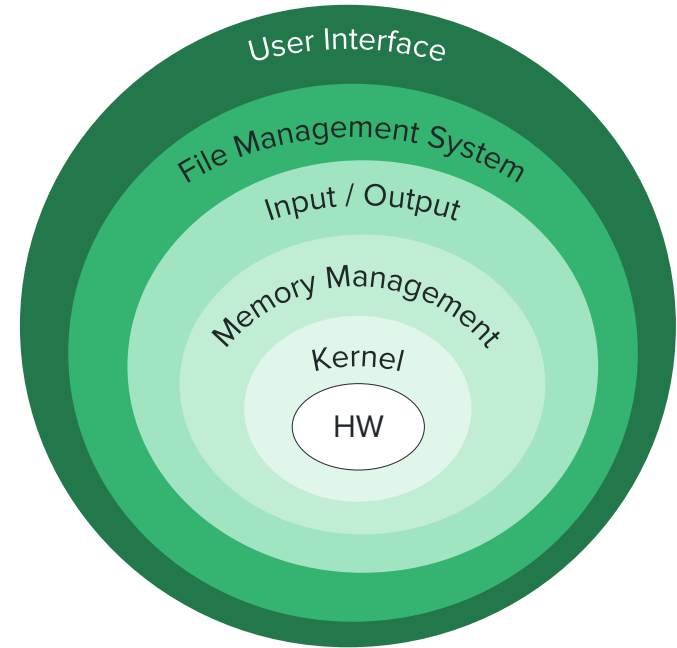


Microkernel  
based Operating System



# Πολυστρωματική προσέγγιση (υβριδικοί πυρήνες)

- Έρχεται να καλύψει τα μειονεκτήματα των μονολιθικών πυρήνων, υιοθετώντας αρχιτεκτονική επιπέδων.
- Κάθε στρώμα (layer) ομαδοποιεί παρόμοιες λειτουργίες
- Οι απαιτήσεις των διεργασιών διαπερνούν αρκετά επίπεδα πριν ολοκληρωθούν.
- Η ρυθμοαπόδοση μπορεί να είναι μικρότερη από τα Λ.Σ. με τους μονολιθικούς πυρήνες.
- Απαιτούνται επιπλέον μέθοδοι για τη μεταβίβαση και τον έλεγχο των δεδομένων.
- Παραδείγματα: NT kernel (windows), macOS kernel



# Μοντέλο πελάτη/διακομιστή

- Λειτουργίες του ΛΣ εκτελούνται από διεργασίες/πελάτες
- Η υλοποίηση του πυρήνα γίνεται σε έναν εξυπηρετητή,
- Ο εξυπηρετητής εκθέτει τις υπηρεσίες του προς τους πελάτες
- Οι πελάτες χρησιμοποιούν κάποιο πρωτόκολλο διαδικεργασικής επικοινωνίας (RPC) για να αιτηθούν μια υπηρεσία από τον πυρήνα
- Ο εξυπηρετητής και οι πελάτες ενδέχεται να βρίσκονται σε διαφορετικά φυσικά μηχανήματα
- Χρησιμοποιείται για την υλοποίηση κατακευμασμένων συστημάτων

# Hypervisor

- Το λογισμικό εικονικοποιεί τους υπολογιστικούς πόρους και τους προσφέρει ως υλικό σε πλήθος εικονικών μηχανών.
- Αποτελεί επιπρόσθετος μηχανισμός αφαίρεσης.
- Κάθε εικονική μηχανή έχει το δικό της ΛΣ και εκτελείται αυτόνομα από τις υπόλοιπες.
- Εικονική μηχανή: εφαρμογή που εξομοιώνει τη λειτουργία ενός υπολογιστικού συστήματος
- Ο hypervisor υλοποιείται ως ΛΣ (Type 1 hypervision) με απευθείας επικοινωνία με το υλικό.
- Ο hypervisor υλοποιείται και ως εφαρμογή (Type 2 hypervision), επικοινωνώντας με τον πυρήνα του υπάρχοντος ΛΣ.