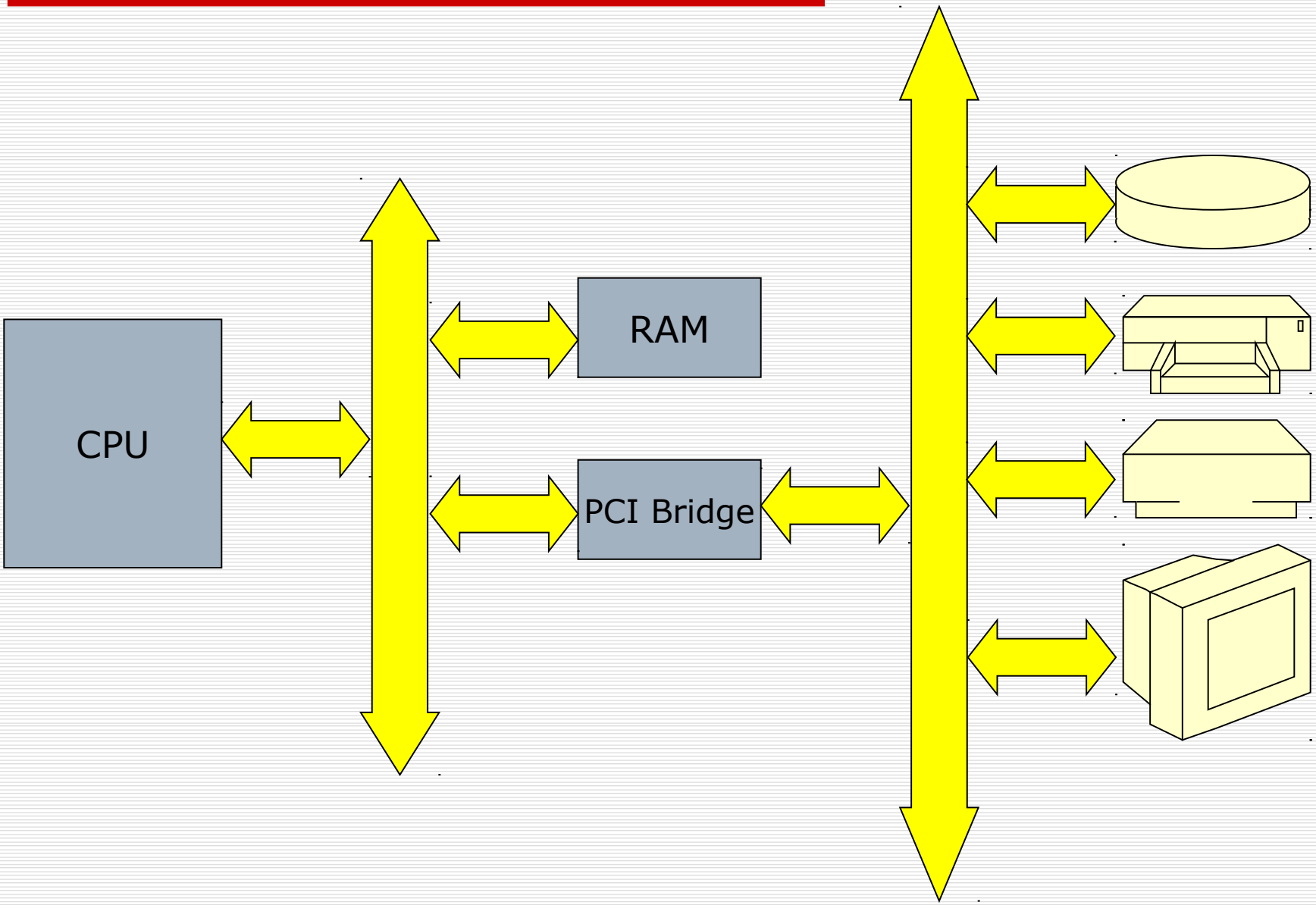
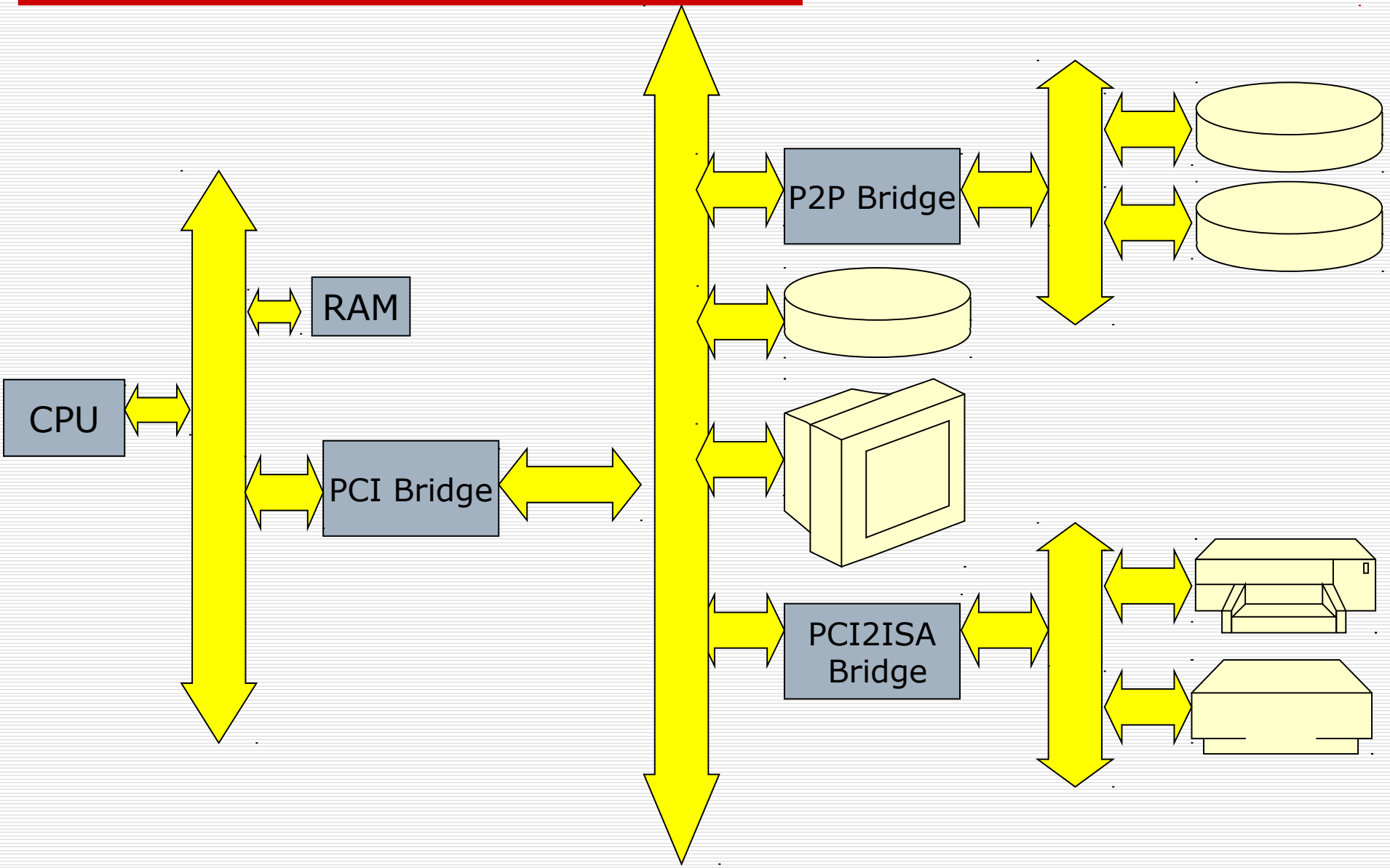
ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΟΡΓΑΝΩΣΗ Η/Υ

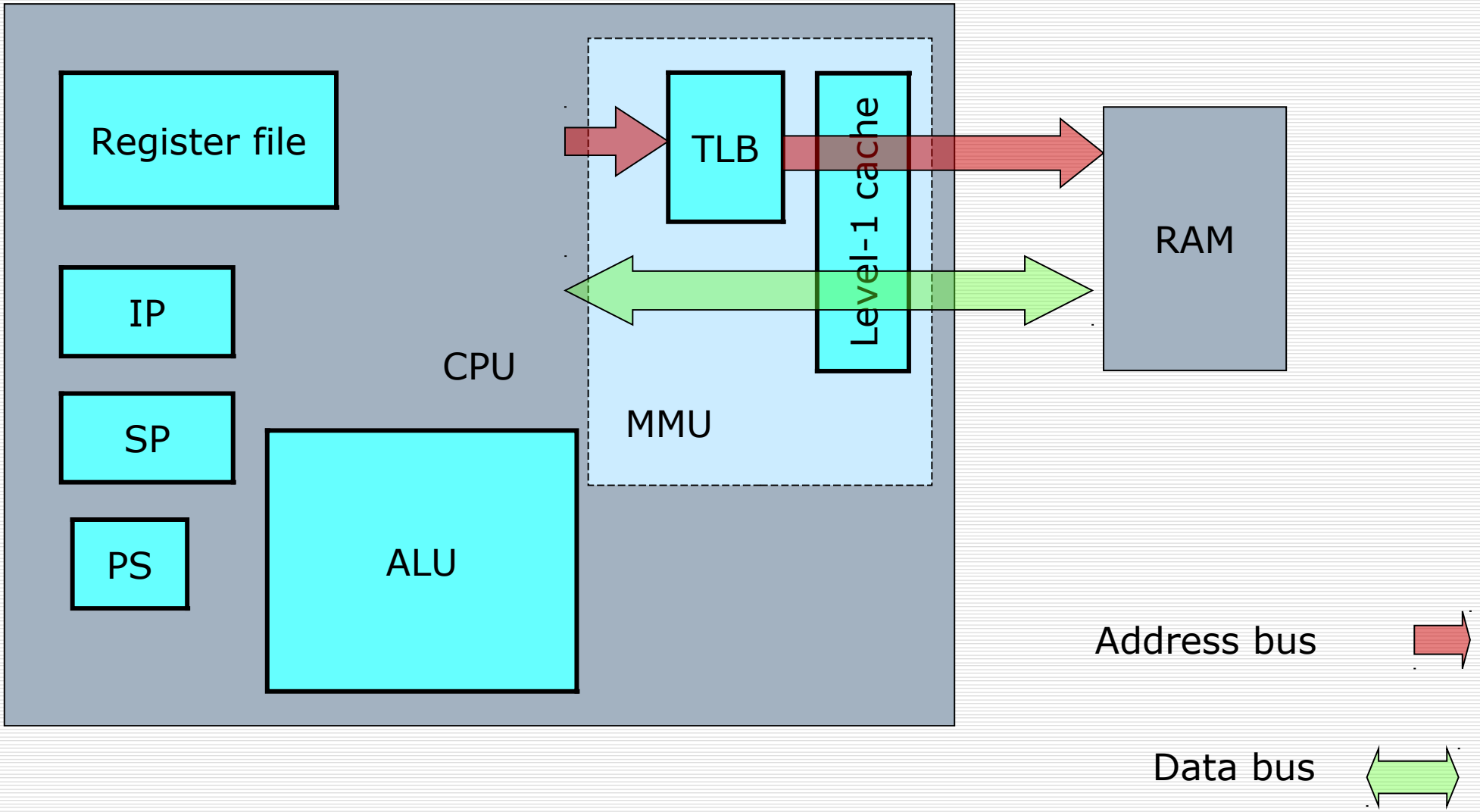
ΔΟΜΗ ΤΟΥ Η/Υ (PC compatibles)



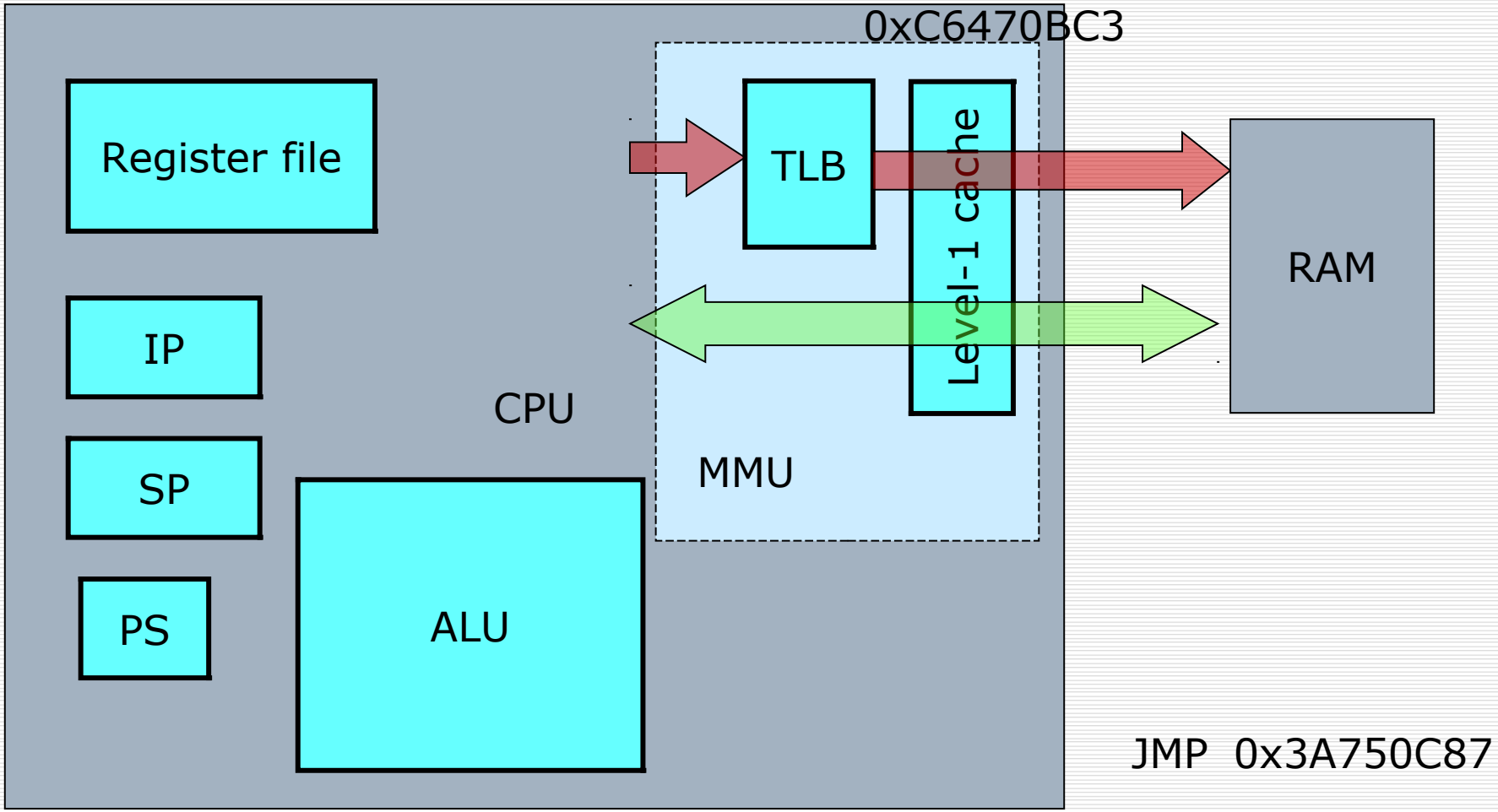
PCI Bridges



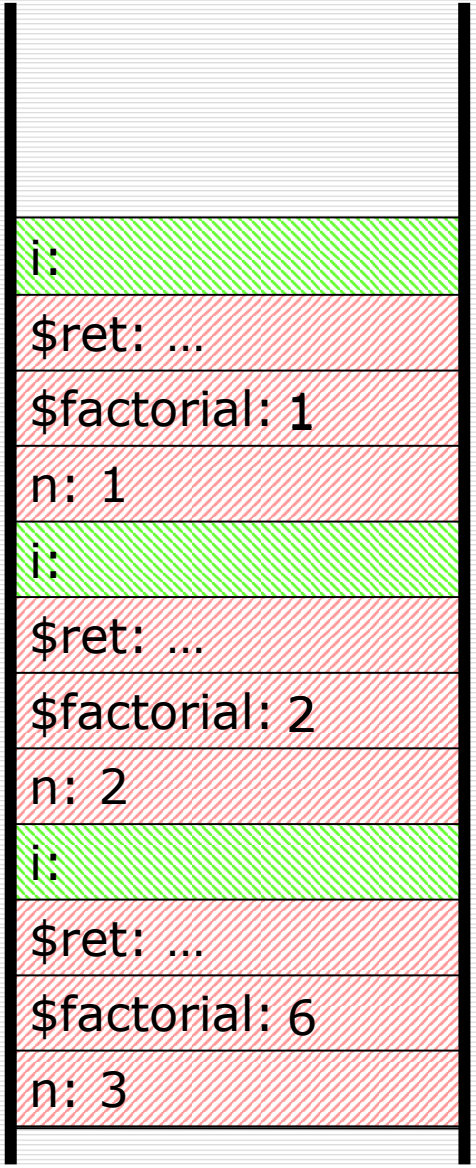
Δομή της CPU


Register file
IP
SP
PS
ALU
CPU
TLB
Level-1 cache
MMU
RAM
Address bus
Data bus

Ιδεατή μνήμη (Virtual memory)



Stack frames

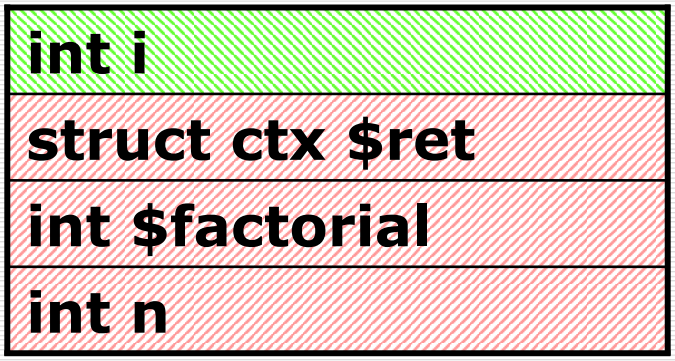


factorial(1)

factorial(2)

factorial(3)

```
int factorial(int n) {
  int i;
  if(n < 2)
    return 1;
  else {
    i = factorial(n-1);
    return n*i;
  }
}
```

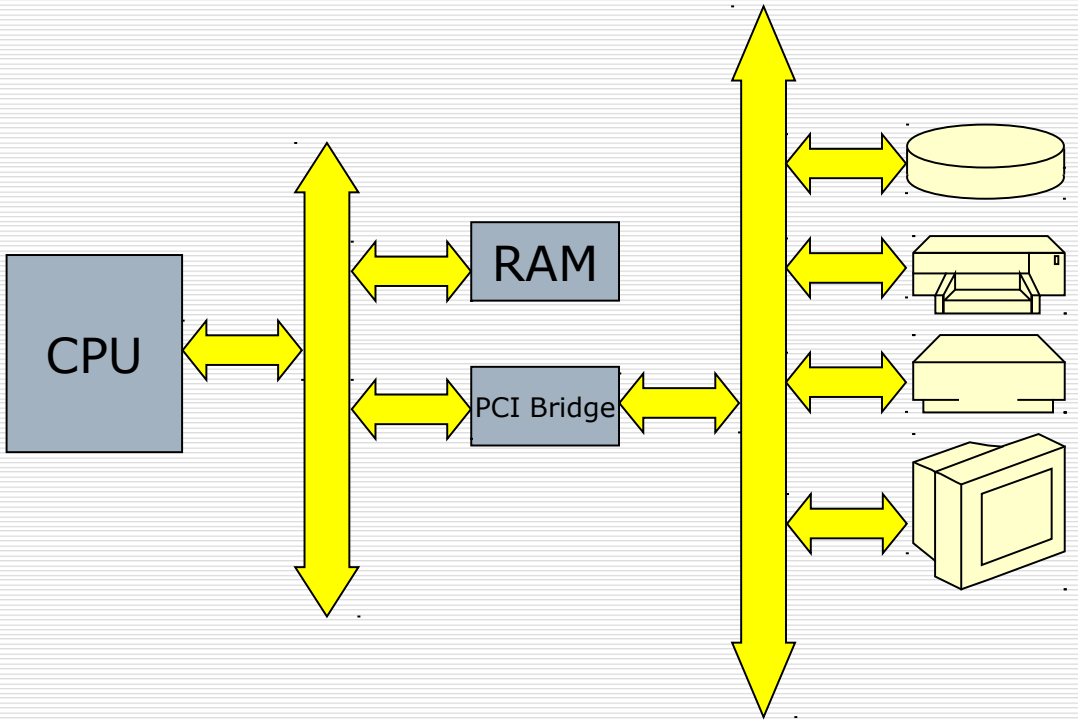


Kernel mode

- Ο επεξεργαστής βρίσκεται σε μια από τις δύο καταστάσεις:
  - **Kernel mode**:
    - Όλες οι εντολές γλώσσας μηχανής είναι νόμιμες.
    - Μόνο ο πυρήνας του ΛΣ πρέπει να εκτελείται σε αυτή την κατάσταση.
  - **User mode**:
    - Κάποιες (οι *προνομιούχες-privileged*) εντολές μηχανής δεν είναι νόμιμες.
    - Τυχόν απόπειρα εκτέλεσης δημιουργεί κατάσταση σφάλματος (θα αναλυθεί αργότερα).
- Η κατάσταση αποθηκεύεται σε ένα bit του PSW register.
- Ο κώδικας των διεργασιών θα πρέπει να εκτελείται μόνο σε User mode, ποτέ σε Kernel mode (λόγοι ασφαλείας).

Σύγχρονες και ασύγχρονες λειτουργίες

- Σύγχρονη λειτουργία (synchronous operation):
  - Αυτός που την ενεργοποιεί, την περιμένει να τελειώσει
  - Πχ. ανάγνωση μιας θέσης μνήμης RAM από τη CPU
- Ασύγχρονη λειτουργία (asynchronous operation):
  - Αυτός που την ενεργοποιεί συνεχίζει χωρίς να την περιμένει να τελειώσει
  - Πχ. ανάγνωση μιας σελίδας από το σκληρό δίσκο
- Πώς γίνεται αντιληπτή η ολοκλήρωση μιας ασύγχρονης λειτουργίας?



Polling and Interrupts

- Polling: το ΛΣ ελέγχει περιοδικά για να δει αν έχει ολοκληρωθεί κάποια λειτουργία.
- Interrupts: όταν ολοκληρωθεί η λειτουργία, η CPU δέχεται ένα *σήμα διακοπής (interrupt signal)*.

- Παραδείγματα:
  - Διάβασμα μιας σελίδας από το σκληρό δίσκο.
  - Ανάγνωση από το πληκτρολόγιο.

Interrupt handling (χειρισμός διακοπής)

- Κάθε διακοπή χαρακτηρίζεται από έναν αριθμό.
- Interrupt vector: ένα array που σε κάθε διακοπή αντιστοιχεί μια διεύθυνση μνήμης, στην οποία βρίσκεται η ρουτίνα χειρισμού διακοπής (interrupt handler).
- Εκτελούνται τα εξής βήματα:
  1. Σώζονται στο σωρό οι καταχωρητές IP και PS.
  2. Η CPU μπαίνει σε kernel mode.
  3. Η εκτέλεση πηδά στον αντίστοιχο interrupt handler.
- Μάσκα διακοπών (interrupt mask): ένα bit vector που δηλώνει για κάθε διακοπή αν αυτή είναι ενεργή (enabled), ή απενεργοποιημένη (disabled).

Trapping

- Για να μπει η εκτέλεση στον πυρήνα, θα πρέπει
  - Να συμβεί κάποιο interrupt, ή
  - Κάποια διεργασία να καλέσει ένα *system call* (*κλήση συστήματος)*.
  - Να συμβεί κάποιο σφάλμα

- Ενιαίος μηχανισμός: κλήση συστήματος = (software) interrupt
- Ονομάζεται *trapping*.

Trapping (συνέχεια)

- Χειρισμός εσφαλμένων καταστάσεων
- Παραδείγματα
  - Απόπειρα εκτέλεσης privileged εντολής σε user mode
  - Απόπειρα πρόσβασης σε απαγορευμένη θέση μνήμης
  - Floating-point error (πχ. division-by-zero, log(0) ...)
  - Εκτέλεση εντολής HALT
  - ... κ.α.

Context switching

- Context: η πληροφορία που χρειάζεται να σωθεί ώστε να συνεχιστεί αργότερα μια διακοπείσα διεργασία.
  - Καταχωρητές
    - Περιλαμβάνονται: IP, SP, …
  - Απεικόνιση ιδεατής μνήμης
- Πώς γίνεται
- Multitasking:
  - Preemptive (με interrupts)
  - Non-preemptive (κλήση `yield`)

Pre-emptive multitasking

- Interrupt-driven.
- Η κάθε διεργασία δρομολογείται για ένα μέγιστο χρόνο, που λέγεται *quantum* (ή *timeslice*).
- Λειτουργία του scheduler:
  - Έστω ότι εκτελείται η διεργασία Α.
  - Έρχεται interrupt από το χρονόμετρο, που ενεργοποιεί τον scheduler.
  - Επιλέγεται η επόμενη διεργασία που θα εκτελεστεί (πώς?), έστω Β.
  - Αρχικοποιείται πάλι το χρονόμετρο για να μετρήσει διάρκεια 1 quantum.
  - `Context_switch(A,B)`

Preemptive multitasking

- Η εκτέλεση μίας διεργασίας που μόλις δρομολογήθηκε κρατάει **το πολύ** 1 quantum.
- Λόγοι που μπορεί να κρατήσει λιγότερο:
  - Η διεργασία ζήτησε την εκτέλεση I/O.
  - Η διεργασία ζήτησε απευθείας (με system call) να σταματήσει η δρομολόγησή της.
- Διάρκεια 1 quantum:
  - Linux: 1 msec – 10 msec
  - Windows NT: 20 - 120 msec
  - Μικρότερο quantum: καλύτερο interactiveness
  - Μεγαλύτερο quantum: καλύτερο locality

Non-preemptive multitasking.

- Χρησιμοποιήθηκε ευρέως στο παρελθόν, σήμερα μόνο σε ειδικές περιπτώσεις (embedded systems).
- Μια διεργασία που εκτελείται δεν διακόπτεται, παρά μόνον αν:
  - Η διεργασία ζητήσει την εκτέλεση I/O.
  - Η διεργασία ζητήσει απευθείας (με system call) να σταματήσει η δρομολόγησή της.
- Δηλ. μπορούμε να θεωρήσουμε ότι

  $$\text{quantum} = \infty$$

APICs

- Advanced Programmable Interrupt Controllers
- Χρησιμοποιούνται για:
  - Προχωρημένο έλεγχο I/O και
  - *κυρίως* Υπολογιστές με πολλαπλά CPU (SMP = Symmetric Multiprocessors)
- Εντολές:
  - Προτεραιότητες: ευέλικτες πολιτικές.
  - Κατεύθυνση του κάθε interrupt σε συγκεκριμένο CPU.
  - Εσωτερικά ρολόγια (timers).
  - ...

Δομή του πυρήνα

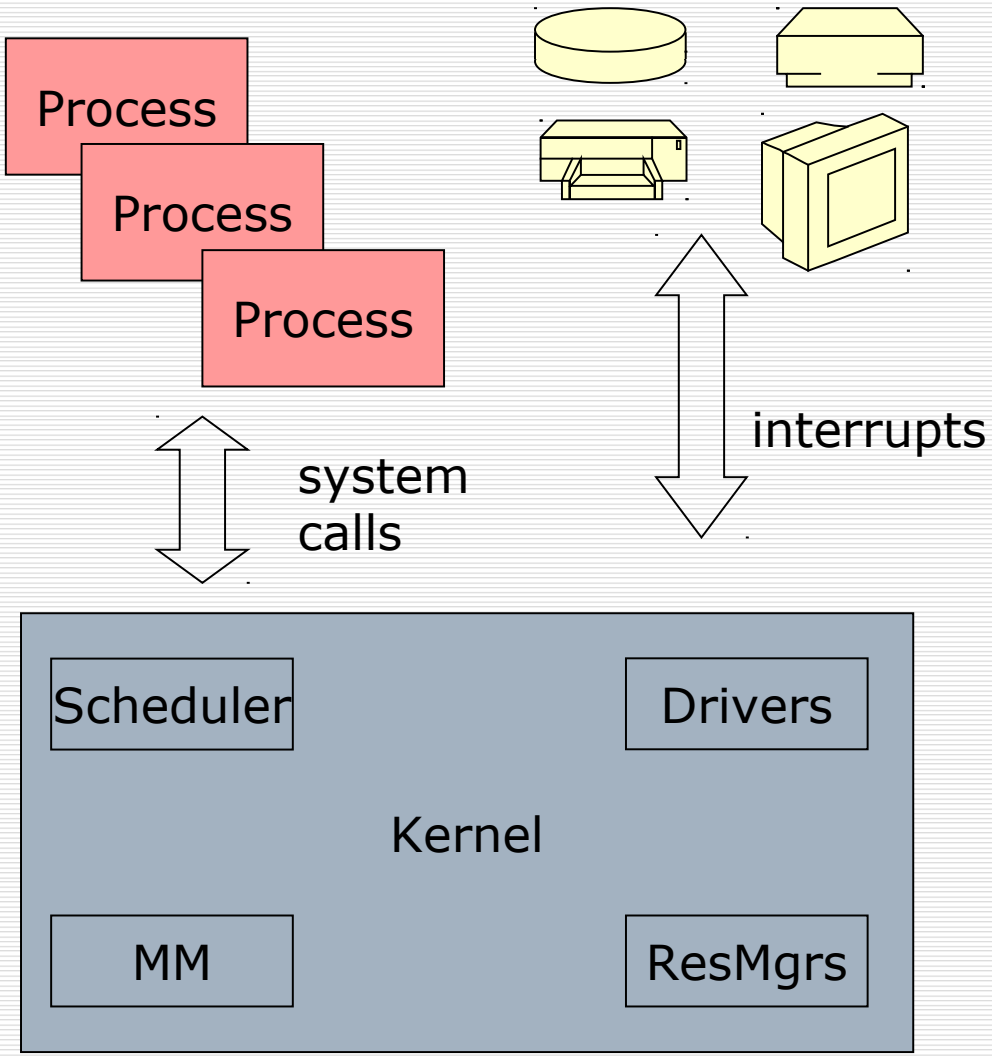
1. Μονολιθική (monolithic): Linux
2. Μικροπυρήνας (microkernel): Windows NT, Solaris
3. Ιδεατές μηχανές (Virtual machines): IBM VM/390

Υπό έρευνα σήμερα:

- Exokernels

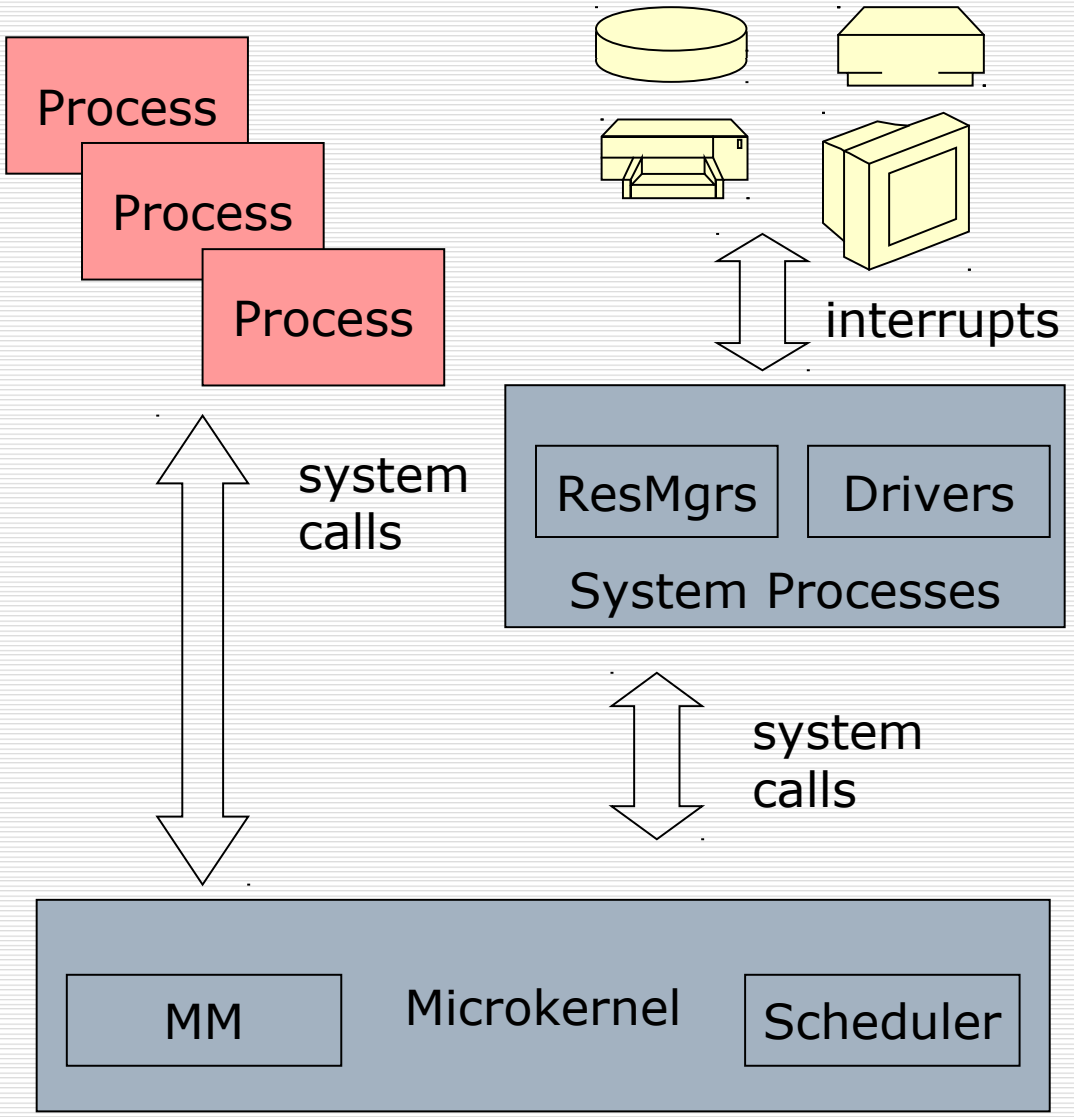
Μονολιθική

- Aka. The big mess
- Ο κώδικας του πυρήνα χωρίζεται σε:
  - Scheduler
  - Memory management
  - Resource managers (filesystem, network protocols)
  - Device drivers
- Είσοδος στον πυρήνα με:
  - Interrupts
  - System calls



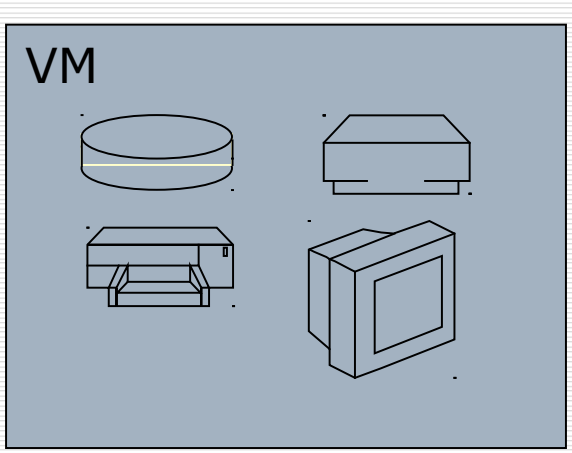
Microkernel

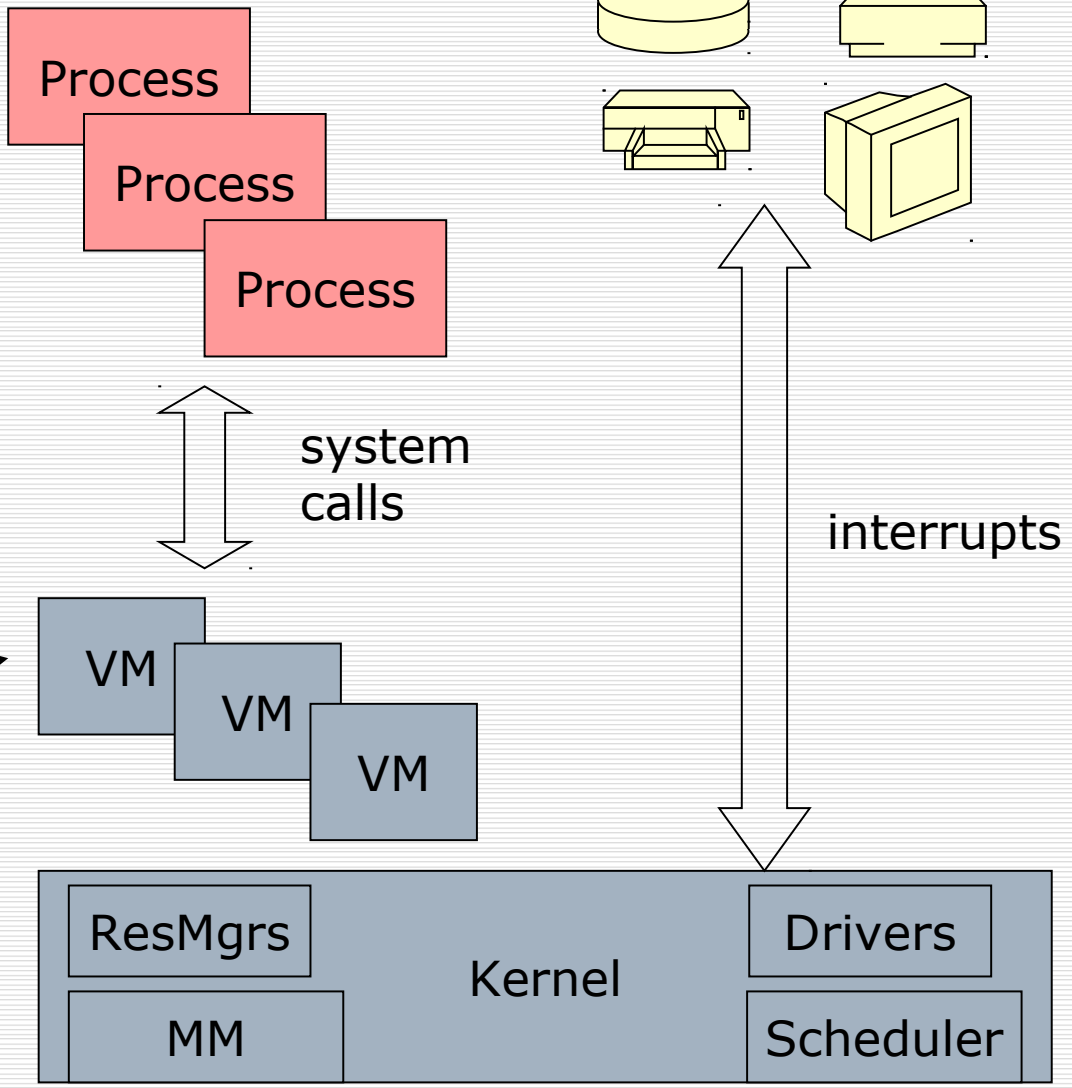
- Ο πυρήνας χωρίζεται σε
  - Μικροπυρήνα
  - Διεργασίες συστήματος (system processes)
- Οι διεργασίες χρήστη ζητούν υπηρεσίες I/O (δίσκου, δικτύου κλπ) από τις διεργασίες συστήματος μέσω μηχανισμών *επικοινωνίας διεργασιών*.



Virtual Machines (VMs)

- Κάθε VM ομοιάζει με έναν πλήρη Η/Υ.
- Δε χρησιμοποιείται σήμερα ως το κυρίως Λ.Σ.
- Υλοποιείται συχνά σε embedded περιβάλλοντα
  - Java VM
  - VMware



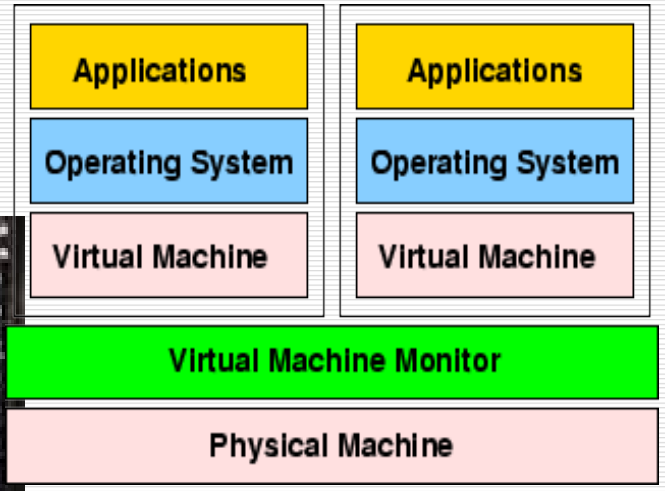


Virtualization

- Σήμερα, έχουμε ένα νέο μοντέλο οργάνωσης:
  - Cluster: συλλογή από (συνήθως όμοιους) υπολογιστές-servers, συνήθως σε κάποιο data cetner
  - Σε κάθε υπολογιστή, ένας αριθμός από VMs.

data center

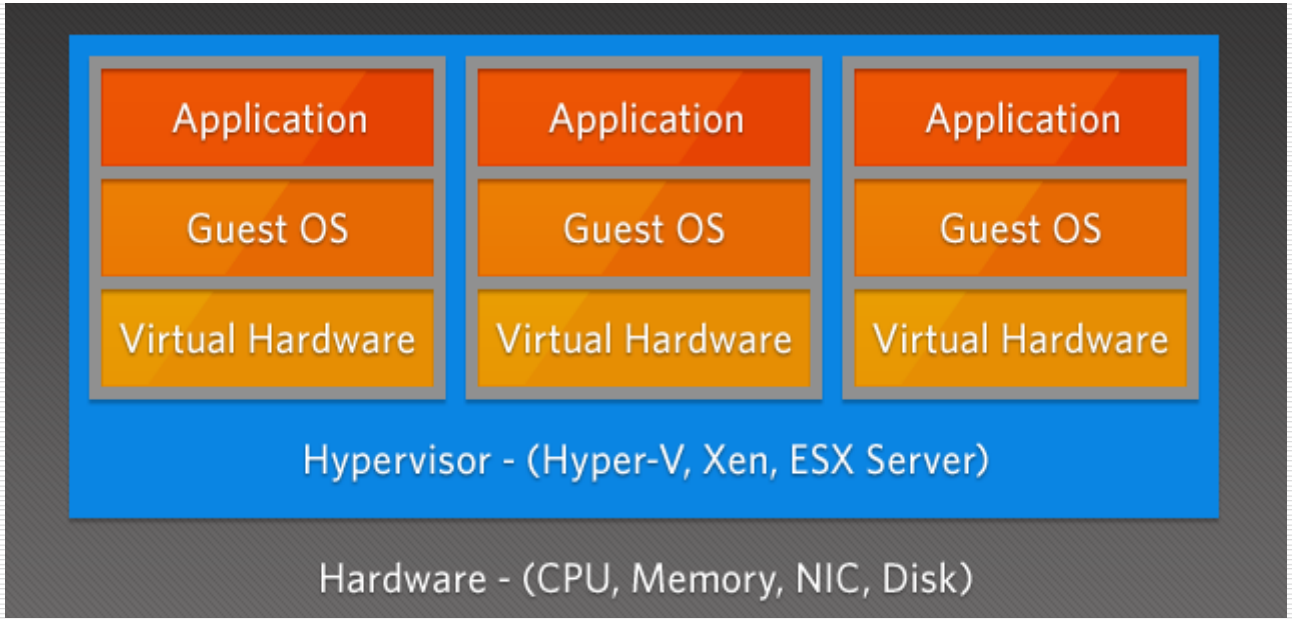
cluster



computer

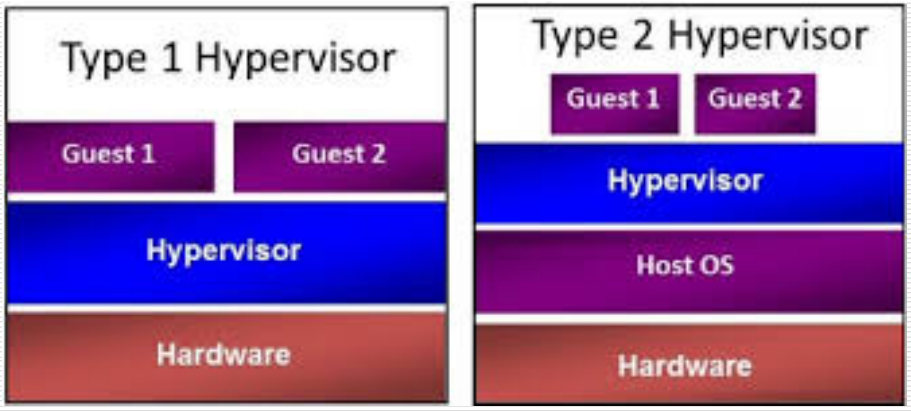
Virtualization techniques

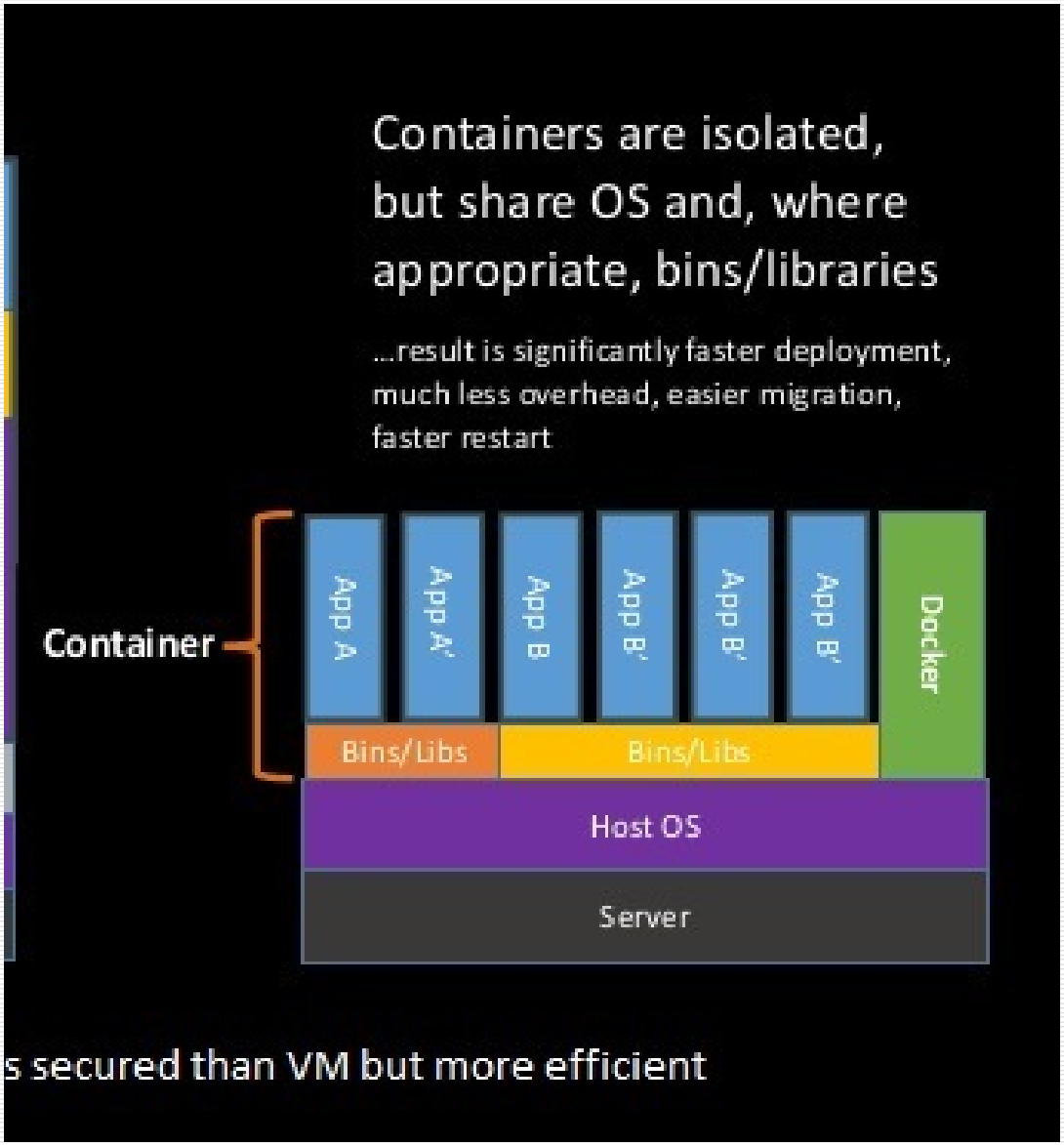
- **Full**: πλήρης προσομοίωση του hardware
  - Εκτελεί guest OS χωρίς αλλαγές
- **Partial**: μερική προσομοίωση του hardware
  - Εκτελεί guest OS με ειδικούς drivers
- **Paravirtualization**: κοινό hardware
  - Guest kernel με αλλαγές, για συνεργασία



Hypervisors (ή VM Monitors)

- Type1 (bare metal)
- Type2 (hosted)
- Container





Cloud computing

- IaaS: Infrastructure as a Service
  - Ενοικίαση VMs
- PaaS: Platform as a Service
  - Περιβάλλον εκτέλεσης εφαρμογών (web server, database, κλπ)
  - Ενοικίαση app hosting
- SaaS: Software as a Service
  - Applications με συνδρομή