



# Λειτουργικά Συστήματα

**Ενότητα 7:** Διαχείριση Μνήμης. Φυσική και Λογική  
Διευθυνσιοδότηση. Προστασία. Εναλλαγή Μνήμης

Δρ. Μηνάς Δασυγένης

[mdasyg@ieee.org](mailto:mdasyg@ieee.org)

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.icte.uowm.gr/mdasyg>

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

---



# Άδειες Χρήσης

---

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Χρηματοδότηση

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



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
*επένδυση στην κοινωνία της γνώσης*  
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
Πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



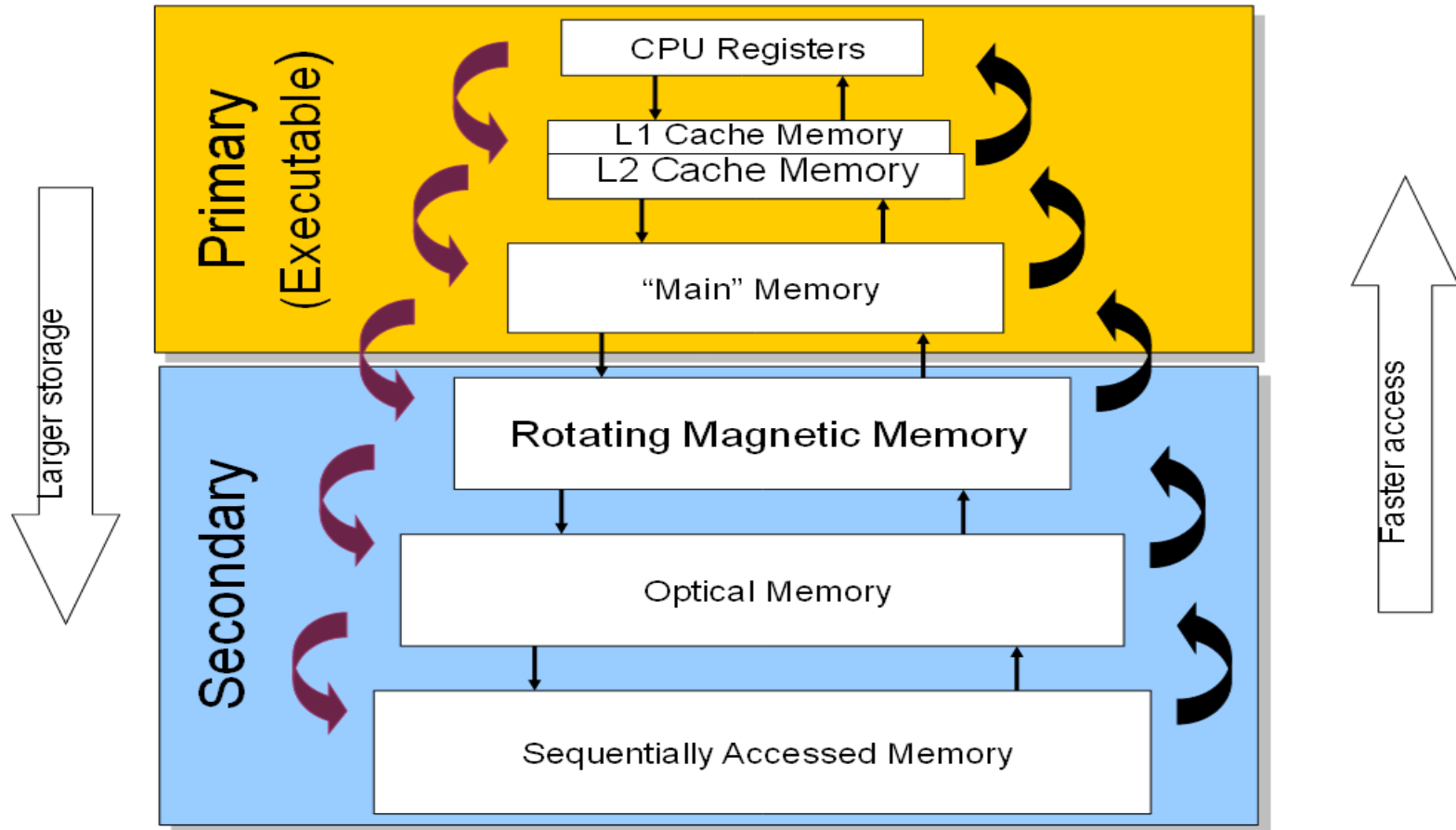
# Σκοπός ενότητας

---

- Η κατανόηση της διαχείρισης μνήμης των ΛΣ.
- Η κατανόηση των λογικών και φυσικών διευθύνσεων μνήμης.



# Όλοι οι σύγχρονοι υπολογιστές διαθέτουν μια ιεραρχία μνήμης



# Ορισμός της Ιεραρχίας Μνήμης

---

Η διαχείριση μνήμης είναι η **λειτουργία της υποδιαίρεσης** της μνήμης από το λειτουργικό σύστημα με δυναμικό τρόπο ώστε να εξυπηρετούνται **όσο το δυνατόν περισσότερες διεργασίες**.



# Ο διαχειριστής μνήμης ανήκει στην καρδιά του κάθε ΛΣ

---

- Τι είναι Διαχειριστής μνήμης;

**Το τμήμα του Λ.Σ. που ασχολείται με την οργάνωση και τις στρατηγικές διαχείρισης της ιεραρχίας μνήμης.**

**Αποστολή του είναι:**

- να παρακολουθεί ποια τμήματα της μνήμης χρησιμοποιούνται και ποια όχι,
- να κατανέμει τη μνήμη στις διεργασίες και να την παίρνει πίσω όταν δεν τη χρειάζονται ή τερματίζουν,
- να διαχειρίζεται την εναλλαγή (Swapping) των διεργασιών μεταξύ κύριας μνήμης και δίσκου.



# Η σημαντικότητα της Διαχείρισης Μνήμης

---

**Είναι απαραίτητη** διότι :

- Η μνήμη είναι ένας **ανεπαρκής πόρος** και είναι απαραίτητη η αποτελεσματική χρήση της.
- Παρέχει **ευκολία** στον προγραμματισμό.
- Ενισχύει τον **πολυπρογραμματισμό**.
- Παρέχει **ασφάλεια και προστασία** στις εκτελούμενες διεργασίες.
- Οι προγραμματιστές επιζητούν την ελαχιστοποίηση του χρόνου προσπέλασης και τη μεγιστοποίηση του μεγέθους της μνήμης για την εκτέλεση των προγραμμάτων (ωστόσο η μνήμη για την εκτέλεση προγραμμάτων πρέπει να είναι cost-effective).





# Απαιτήσεις μνήμης των Microsoft Windows

---

<i>Operating System</i>	<i>Release Date</i>	<i>Minimum Memory Requirement</i>	<i>Recommended Memory</i>
Windows 1.0	November 1985	256KB	
Windows 2.03	November 1987	320KB	
Windows 3.0	March 1990	896KB	1MB
Windows 3.1	April 1992	2.6MB	4MB
Windows 95	August 1995	8MB	16MB
Windows NT 4.0	August 1996	32MB	96MB
Windows 98	June 1998	24MB	64MB
Windows ME	September 2000	32MB	128MB
Windows 2000 Professional	February 2000	64MB	128MB
Windows XP Home	October 2001	64MB	128MB
Windows XP Professional	October 2001	128MB	256MB

**Κάθε νέα έκδοση αυξάνει τις ελάχιστες  
απαιτήσεις σε μνήμη!**



# 4 είναι τα χαρακτηριστικά των Διαχειριστών Μνήμης

---

- **Εκχωρούν** την πρωτεύουσα μνήμη σε διεργασίες.
- **Αντιστοιχούν** το χώρο διευθύνσεων της διεργασίας στην κύρια μνήμη.
- **Ελαχιστοποιούν** το χρόνο προσπέλασης χρησιμοποιώντας cost-effective τεχνικές, στατικές ή δυναμικές.
- **Αλληλεπιδρούν** με ειδικό hardware για τη διαχείριση της μνήμης (MMU) για να βελτιώσουν την απόδοση.



# Στρατηγικές διαχείρισης μνήμης

---

Σχεδιάζονται έτσι ώστε να είναι εφικτή η βέλτιστη δυνατή χρήση της κύριας μνήμης.

Διακρίνονται σε :

- **Στρατηγικές προσκόμισης:**

Καθορίζουν το σημείο όπου θα τοποθετηθεί το επόμενο τμήμα προγράμματος ή δεδομένων, καθώς μετακινείται από τη δευτερεύουσα μνήμη (δηλ. από το δίσκο).

- **Στρατηγικές τοποθέτησης:**

Καθορίζουν το σημείο της κυρίας μνήμης όπου το σύστημα θα μπορούσε να τοποθετήσει τμήματα δεδομένων.

- **Στρατηγικές επανατοποθέτησης:**

Καθορίζουν ποιο τμήμα θα αφαιρεθεί από την κύρια μνήμη στις περιπτώσεις όπου η κύρια μνήμη είναι αρκετά πλήρης ώστε να παρέχει χώρο σε ένα νέο πρόγραμμα.



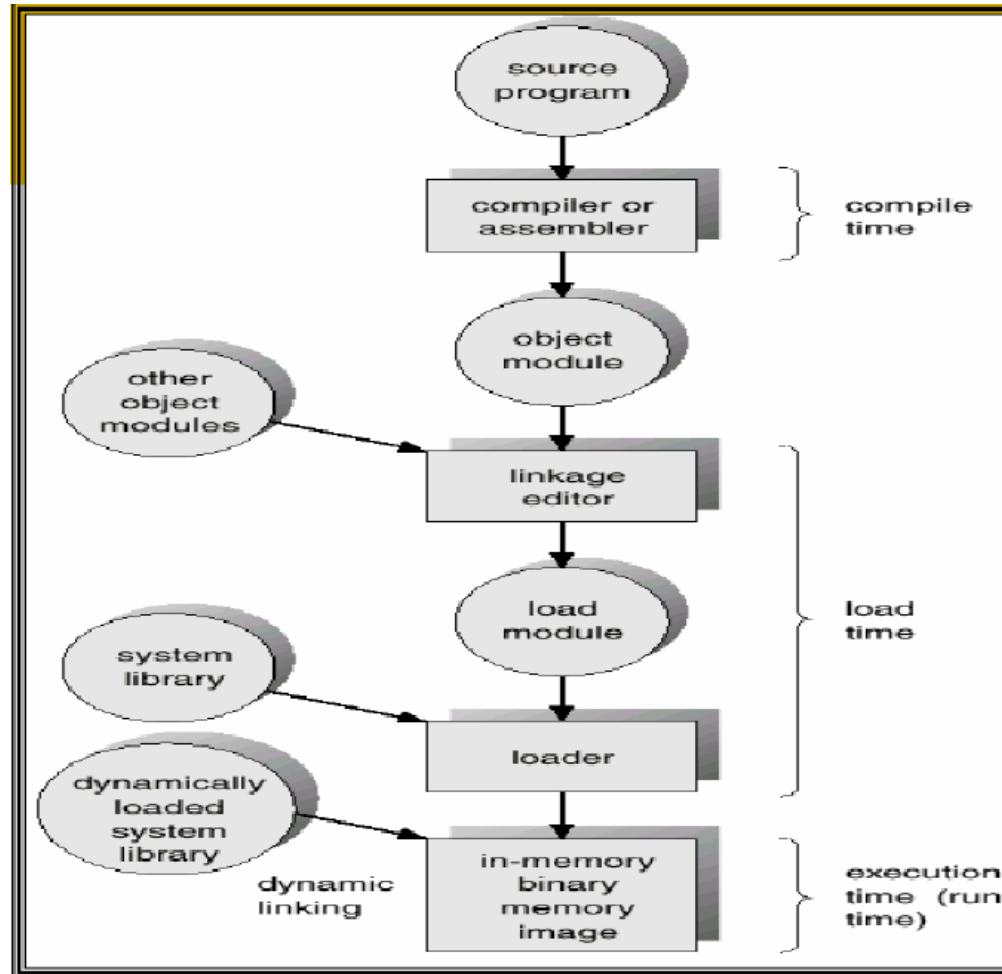
# Η δέσμευση των διευθύνσεων μνήμης γίνεται σε 3 φάσεις

---

- **Χρόνος μετάφρασης:** εάν η θέση μνήμης είναι γνωστή εκ των προτέρων, μπορεί να δημιουργηθεί απόλυτος κώδικας. Πρέπει να επαναληφθεί η μετάφραση εάν αλλάξει η αρχική θέση (δε χρησιμοποιείται).
- **Χρόνος φόρτωσης:** εάν η θέση μνήμης δεν είναι γνωστή κατά το χρόνο μετάφρασης, πρέπει να δημιουργηθεί επανατοποθετήσιμος κώδικας.
- **Χρόνος εκτέλεσης:** η δέσμευση καθυστερεί μέχρι τον χρόνο εκτέλεσης εάν η διεργασία μπορεί να μετακινηθεί κατά την εκτέλεση, από ένα τμήμα μνήμης σε κάποιο άλλο. Απαιτεί υποστήριξη από το υλικό για χαρτογράφηση των διευθύνσεων.



# Δέσμευση Διευθύνσεων μιας διεργασίας



# Συνεχόμενη και μη συνεχόμενη εκχώρηση μνήμης

---

- Η **συνεχόμενη εκχώρηση μνήμης** αφορά τα πρώτα υπολογιστικά συστήματα όπου αν το πρόγραμμα ήταν μεγαλύτερο από τη διαθέσιμη μνήμη το σύστημα δεν μπορούσε να το εκτελέσει.
- Στη **μη συνεχόμενη εκχώρηση μνήμης** το πρόγραμμα διαιρείται σε τεμάχια ή τμήματα που τοποθετούνται από το σύστημα σε μη γειτονικές σχισμές στην κύρια μνήμη. Η τεχνική αυτή κάνει εφικτή τη χρήση περιοχών που είναι πολύ μικρές για να χωρέσουν ολόκληρο πρόγραμμα. Αν και με τον τρόπο αυτό εισάγεται στο σύστημα πολυπλοκότητα αυτή δικαιολογείται από την αύξηση που επιτυγχάνεται στο βαθμό πολυπρογραμματισμού.



# Βασική διαχείριση μνήμης

---

- **Μονοπρογραμματισμός:**

Ένας χρήστης μονοπωλεί τη χρήση του συστήματος και όλοι οι πόροι είναι αφιερωμένοι σ' αυτόν.

- **Επικαλύψεις (overlays):**

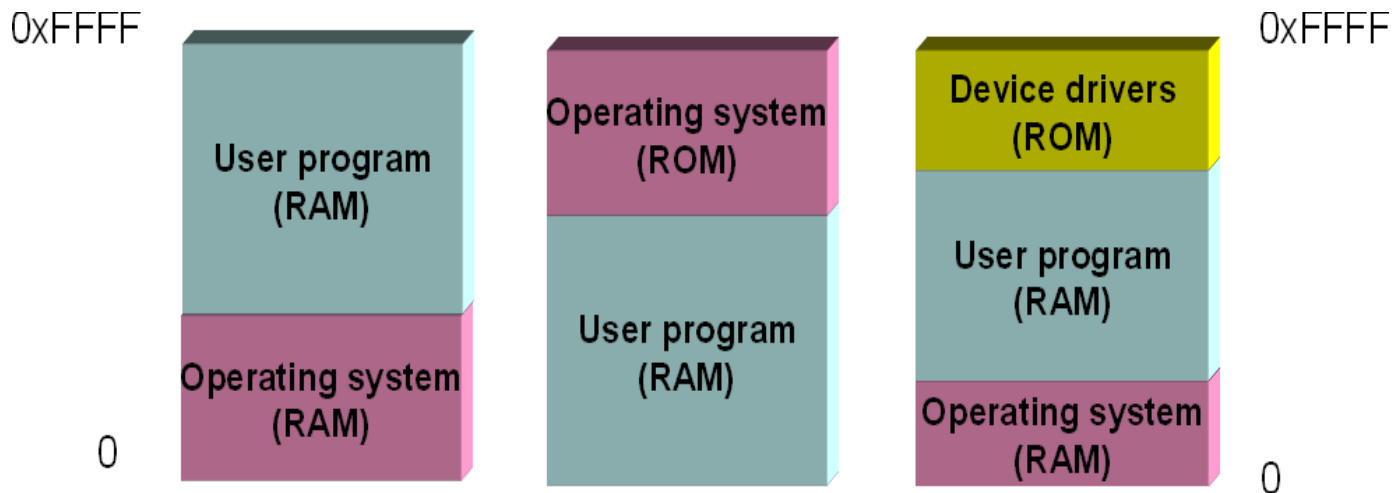
Τεχνική που επιτρέπει σε ένα σύστημα να εκτελεί προγράμματα που είναι μεγαλύτερα από την κύρια μνήμη. Ο προγραμματιστής διαιρεί το πρόγραμμα σε λογικές ενότητες. Όταν το πρόγραμμα δεν χρειάζεται μνήμη για ένα τμήμα, το σύστημα μπορεί να αντικαταστήσει όλη ή μέρη της κύριας μνήμης για να καλύψει μια ανάγκη (δηλαδή, να φορτώσει μια άλλη ενότητα).



# Μονοπρογραμματισμός

Τρεις απλοί τρόποι οργάνωσης μνήμης σε Λ.Σ. με μια διεργασία χρήστη.

Η προσασία μνήμης δεν τίθεται ως πρόβλημα (Μόνο μια διεργασία υπάρχει).



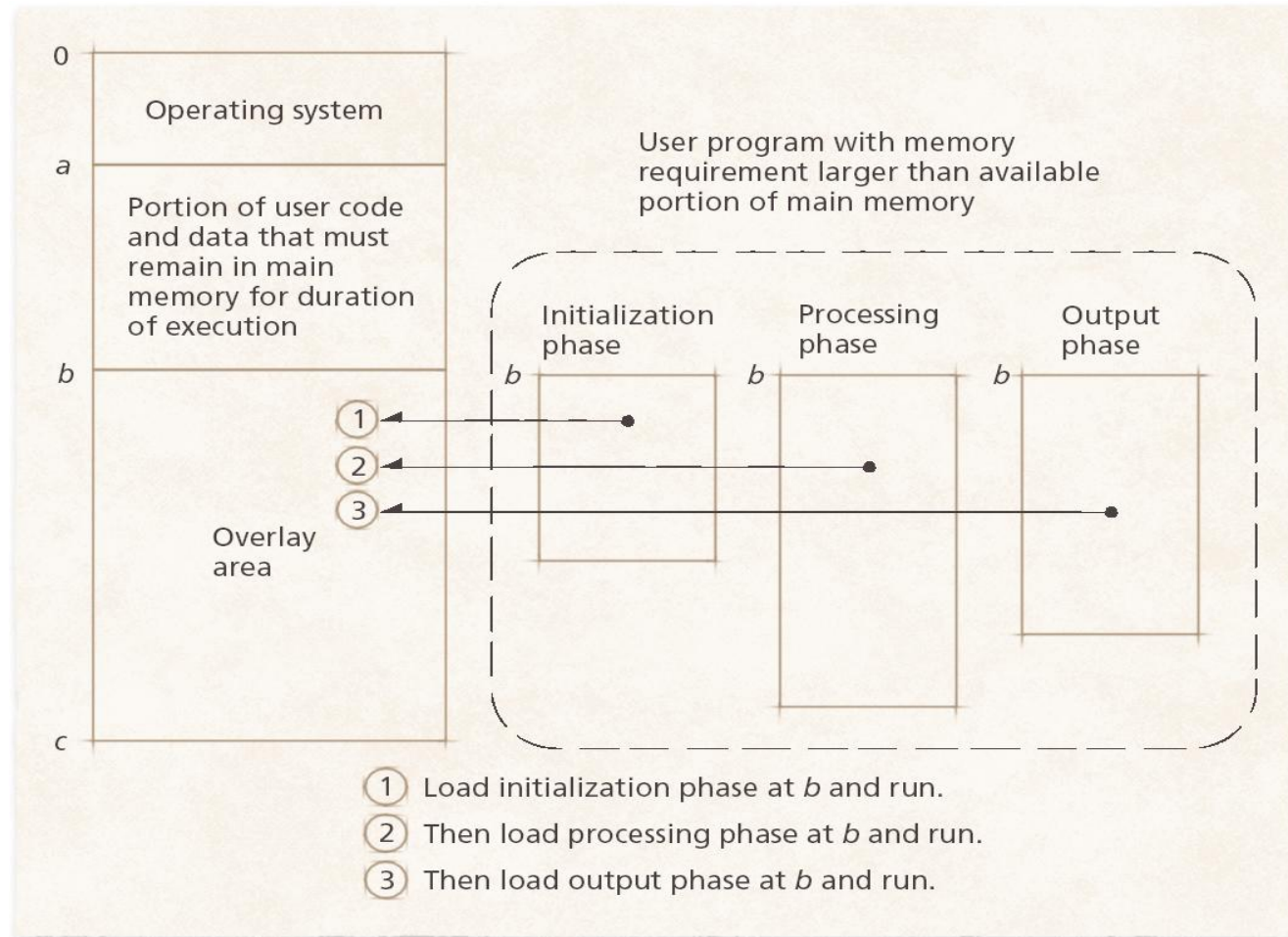


# Επικαλύψεις ή διαστρωματώσεις ή overlays (1/2)

- Διατηρούμε στη μνήμη μόνο τις εντολές και τα δεδομένα εκείνα που απαιτούνται σε κάθε χρονική στιγμή.
- Χρειάζεται όταν η διεργασία είναι μεγαλύτερη από το χώρο μνήμης που της έχει ανατεθεί.
- Υλοποιείται από τον χρήστη, δεν χρειάζεται ιδιαίτερη υποστήριξη από το Λ.Σ.



# Επικαλύψεις ή διαστρωματώσεις ή overlays (2/2)



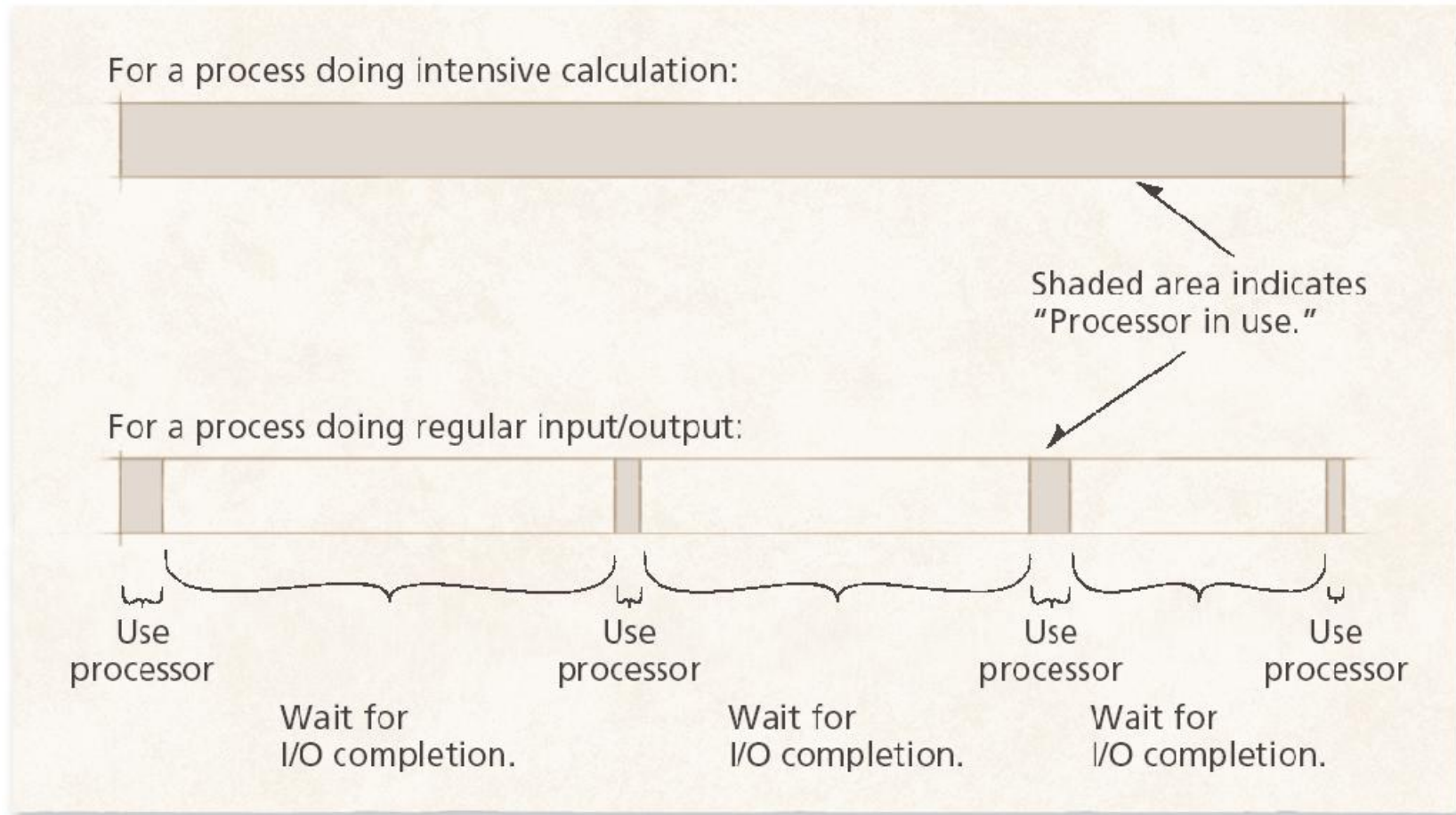
# Ο πολυπρογραμματισμός αυξάνει το ποσοστό χρήσης της CPU

---

- Η χρήση του επεξεργαστή από μία διεργασία διακόπτεται συχνά, λόγω της ανάγκης για **λειτουργίες I/O** που είναι υπερβολικά αργές συγκρινόμενες με την ταχύτητα της CPU.
- Η αύξηση της χρήσης της CPU επιτυγχάνεται με τα συστήματα πολυπρογραμματισμού, όπου **αρκετοί χρήστες ανταγωνίζονται συγχρόνως για τους πόρους του συστήματος**. Έτσι αρκετές διεργασίες πρέπει να βρίσκονται στην κύρια μνήμη την ίδια στιγμή, ώστε αν κάποια υλοποιεί λειτουργίες I/O, κάποια άλλη να χρησιμοποιεί την CPU, ώστε να αυξάνεται το ποσοστό χρήσης της CPU και η απόδοση (throughput) του συστήματος.



# Χρήση της CPU σε σύστημα ενός χρήστη: Μεγάλα IDLE Διαστήματα



# Πόσα προγράμματα είναι αρκετά για μια CPU ;

---

Για τον καθορισμό του πλήθους των διεργασιών που μπορούν να υπάρχουν συγχρόνως στην κύρια μνήμη πρέπει να ληφθούν υπόψη και να εξισορροπηθούν τα εξής:

- Περισσότερες διεργασίες χρησιμοποιούν καλύτερα την CPU, αλλά απαιτείται καλύτερη διαχείριση και προστασία της μνήμης.
- Λιγότερες διεργασίες χρησιμοποιούν λιγότερη μνήμη (φθηνότερα!).
- Περισσότερη αναμονή για I/O σημαίνει μικρότερη χρήση επεξεργαστή.



# Δυο είναι τα θεμελιώδη προβλήματα του Πολυπρογραμματισμού

---

- **Επανατοποθέτηση** (relocation) (πιθανή τοποθέτηση διεργασιών σε διαφορετική διεύθυνση μνήμης κατά τη φόρτωση).
- **Προστασία** (protection) (ανάγνωση και εγγραφή τμημάτων μνήμης που ανήκουν σε άλλες διεργασίες ή χρήστες).



# Η μνήμη διαιρείται σε λογικό και φυσικό χώρο διευθύνσεων

---

- «**Λογική διεύθυνση**», είναι η διεύθυνση μνήμης που δημιουργείται από τον επεξεργαστή και είναι η διεύθυνση που γνωρίζει και χρησιμοποιεί η διεργασία. Ονομάζεται και ιδεατή διεύθυνση (virtual address).
- «**Φυσική διεύθυνση**», είναι η διεύθυνση που αντιστοιχεί σε μια πραγματική διεύθυνση μνήμης (hardware).
  - Κατά το χρόνο μετάφρασης ή εκτέλεσης, η λογική διεύθυνση αντιστοιχίζεται σε πραγματική διεύθυνση.



# Η Μονάδα Διαχείρισης Μνήμης επιτελεί την αντιστοίχιση ιδεατή $\Leftrightarrow$ φυσική διεύθυνση

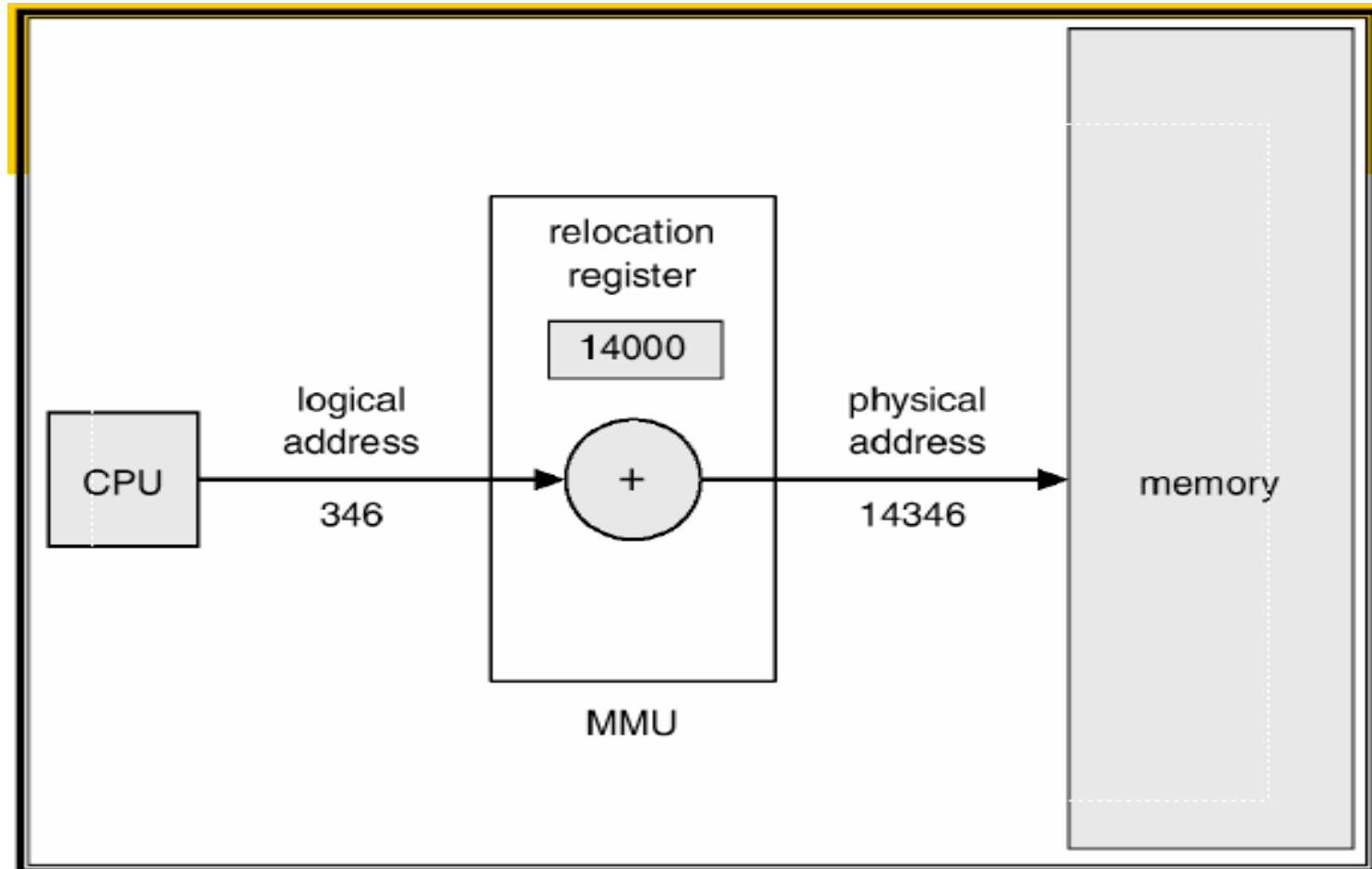
---

- Φυσική συσκευή (real hardware) που χαρτογραφεί ιδεατές σε φυσικές διευθύνσεις.
- Σε ένα MMU σχήμα, η τιμή του καταχωρητή επανατοποθέτησης προστίθεται σε κάθε διεύθυνση που δημιουργείται από μία διεργασία χρήστη κατά τη χρονική στιγμή που αποστέλλεται στη μνήμη.
- Το πρόγραμμα χρήστη αφορά σε ιδεατές διευθύνσεις, δεν χρειάζεται ποτέ να δει τις πραγματικές φυσικές διευθύνσεις.

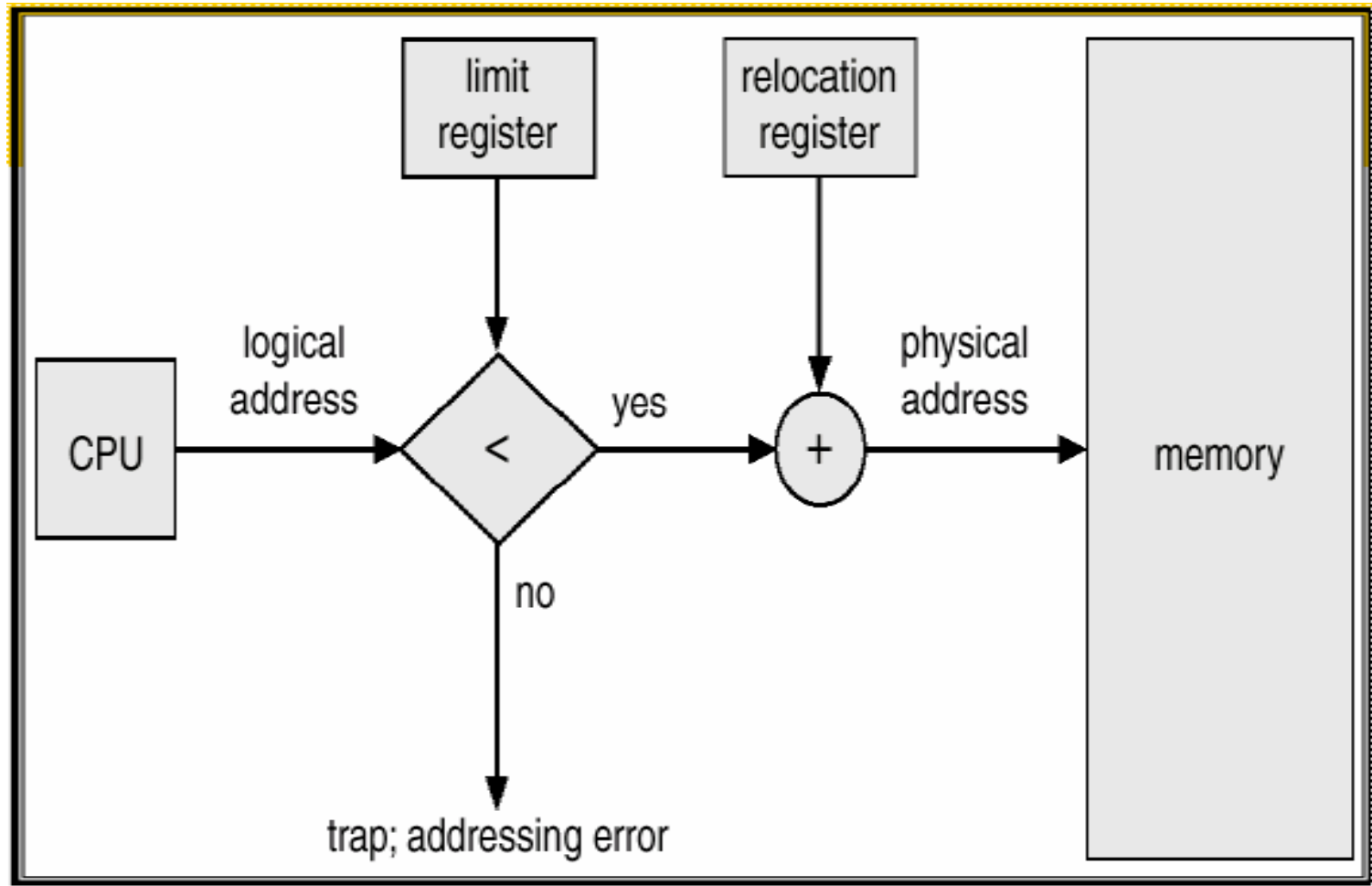




# Παράδειγμα MMU



# Παράδειγμα MMU με προστασία



# Εικονική διεύθυνση των windows

---

- Όλες οι διεργασίες (32bit) βλέπουν μια μνήμη από 0 έως 4,294,967,295 (4GB).
- Από αυτά τα 2GB είναι ιδιωτική μνήμη της διεργασίας και τα υπόλοιπα 2GB είναι του ΛΣ.
- Ασφαλώς δε χρησιμοποιούνται και τα 2GB από κάθε διεργασία.
- Κάθε διεργασία βλέπει τα δικά της 2GB (ιδιωτική μνήμη).
  - Για παράδειγμα η διεύθυνση 0x100 της διεργασίας A είναι διαφορετική από τη διεύθυνση 0x100 της διεργασίας B.



# Τι είναι το PAE (physical address extension) (1/2)

---

- Είναι μια τεχνική για να αυξηθεί η δυνατότητα διευθυνσιοδότησης μνήμης στα x86 (32bit) συστήματα ώστε να υποστηρίζεται μνήμη πέρα από 4GB.
- Υποστηρίζεται από το Pentium (1995) και ύστερα (απαιτούνται έξτρα κανάλια στο addr-bus).
- Δε χρειάζεται στα συστήματα 64bit.
- Δεν αλλάζει την ιδιωτική μνήμη που έχει κάθε διεργασία, μόνο το ποσό της μνήμης που μπορεί να χρησιμοποιήσει ο επεξεργαστής.
- Γίνεται μετατροπή από διευθύνσεις μνήμης 32bit (4GB max) σε 36bit (64GB max) έως 52(bit).



# Τι είναι το PAE

## (physical address extension) (2/2)

---

- Απαιτείται υποστήριξη από **hardware** (cpu) και **software** (OS).
- Στο linux/FreeBSD (32bit) μπορεί να γίνει compile ως παράμετρος του πυρήνα του ΛΣ.
- Στα windows (32bit) η Microsoft έχει κλειδώσει το PAE να μη χρησιμοποιείται και αναγκαστικά πρέπει κάποιος να αγοράσει windows 64bit.
- Το PAE δεν επιτρέπει κάθε διεργασία να βλέπει 64GB. Οι 32bit διεργασίες βλέπουν 4GB χώρο διευθύνσεων. Το ΛΣ όμως βλέπει όλα τα 64GB.
- Το PAE εισάγει μια (μικρή) καθυστέρηση στην πρόσβαση στη μνήμη, γιατί χρησιμοποιούμε καταχωρητές 32bit για να διεθυσιοδοτήσουμε παραπάνω bit.



# Πως χρησιμοποιείται η μνήμη πάνω από 4 GB σε 32bit ΛΣ

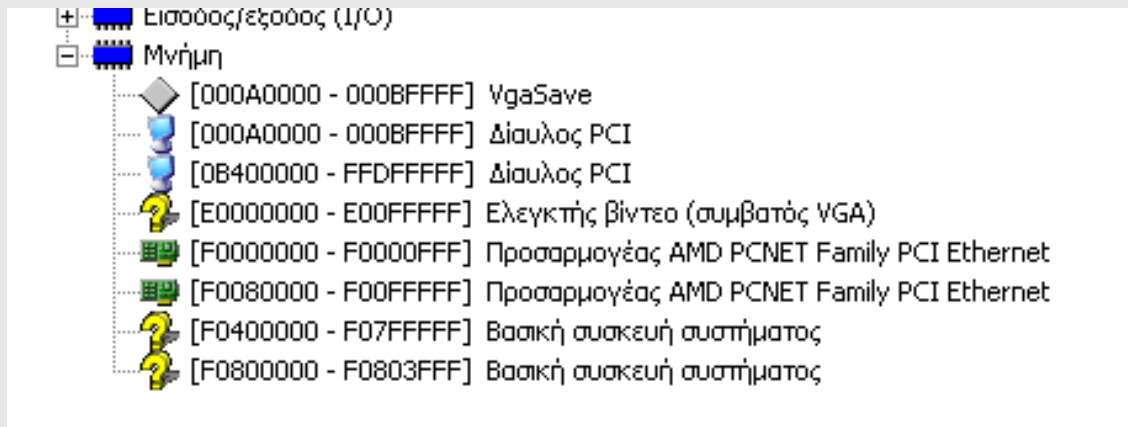
- Η επιλογή για μνήμη μεγαλύτερη από 4GB είναι λοιπόν:
  - 32bit ΛΣ με PAE.
  - 64bit ΛΣ.
- Σε σύγκριση ταχύτητας πρόσβασης στη μνήμη το 32bit ΛΣ είχε δείκτη windows 5.1, ενώ το 64bit ΛΣ είχε δείκτη windows 5.0.

**==> Όμως ακόμη και αν το σύστημα έχει λιγότερα από 4GB μπορεί να χρειάζεται το PAE (επόμενη διαφάνεια).**



# Γιατί τα 4GB χωρίς ΡΑΕ φαίνονται ως 3GB;

- Οι περισσότερες συσκευές χρησιμοποιούν διευθύνσεις μνήμης ως είσοδο-έξοδο. Αυτό μπορεί να φανεί στα windows από τη διαχείριση συσκευών.



- Επειδή αυτές οι διευθύνσεις μνήμης των συσκευών πέφτουν πάνω σε διευθύνσεις μνήμης RAM, τα αντίστοιχα τμήματα της RAM είναι μη διαθέσιμα. Έτσι αν έχουμε 4GB π.χ. δε θα μπορέσουμε να τα χρησιμοποιήσουμε όλα.

# Γιατί απαιτείται η επαναχαρτογράφηση (remapping); Ποιο πρόβλημα επιλύει;

---

- Μερικές συσκευές χρησιμοποιούν διευθύνσεις μνήμης για είσοδο έξοδο και έτσι η μνήμη αυτών των διευθύνσεων δε μπορεί να χρησιμοποιηθεί.
- Για να χρησιμοποιηθούν αυτές οι διευθύνσεις το BIOS επαναχαρτογραφεί αυτά τα τμήματα σε υψηλότερες διευθύνσεις μη χρησιμοποιούμενες.
- Όμως αν χαρτογραφηθούν σε διευθύνσεις πάνω από 4GB, τότε δε θα χρησιμοποιηθούν από το ΛΣ αν δεν έχει PAE ή αν δεν είναι 64bit.
- Η λειτουργία remapping γίνεται από το BIOS και το chipset.

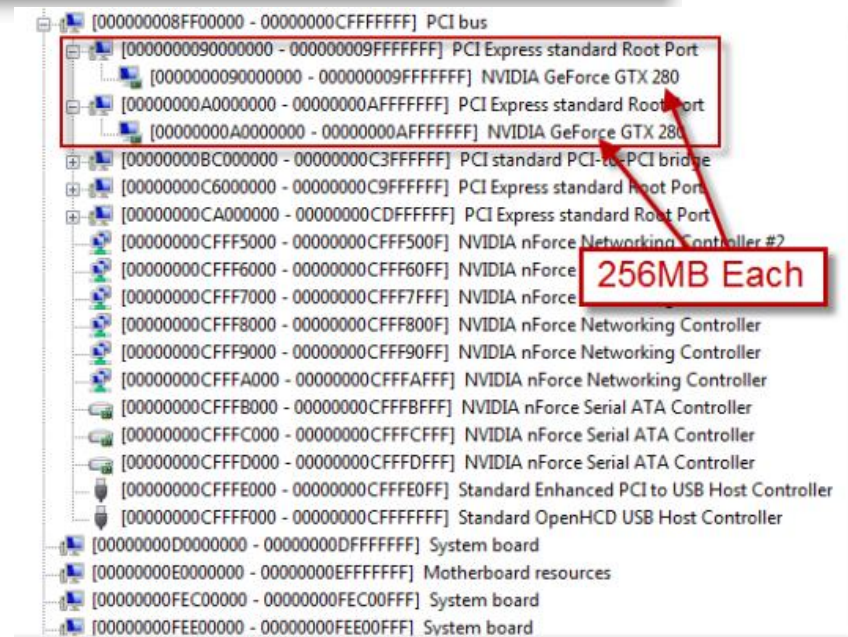




# Κάποιες συσκευές χρησιμοποιούν διευθύνσεις μνήμης

```
Administrator: Command Prompt
C:\>meminfo -r
MemInfo v1.11 - Show PFN database information
Copyright (C) 2007-2008 Alex Ionescu
www.alex-ionescu.com
Physical Memory Range: 0000000000001000 to 0000000000009B000 <154 pages, 616 KB>
Physical Memory Range: 00000000000100000 to 000000008FEF0000 <589296 pages, 2357184 KB>
Physical Memory Range: 00000000100000000 to 0000000170000000 <458752 pages, 1835008 KB>
MmHighestPhysicalPage: 1507328
```

**9F0000 to page 100000  
(κενό)**



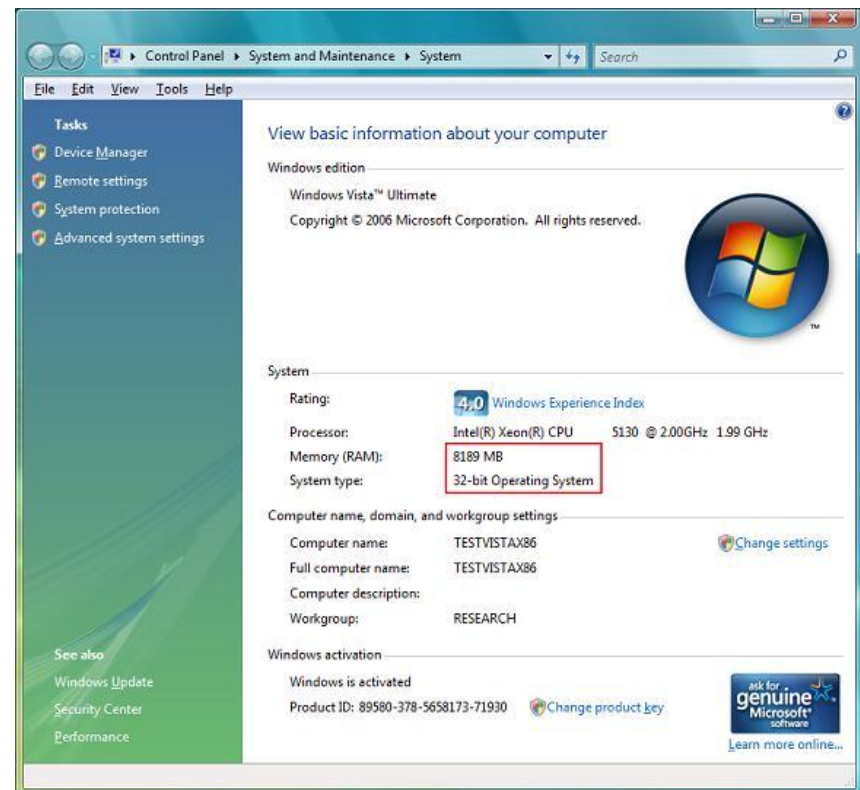
TIP1: Αν έχετε ΛΣ 32bit μην αγοράσετε κάρτα γραφικών με μεγάλο ποσό μνήμης!!!

TIP2: Αν στο BIOS έχει επιλογή memory-remap να την ενεργοποιήσετε!



# PAE σε 32bit windows vista

- Με κατάλληλη επέμβαση στο registry ενεργοποιείται το PAE σε 32bit Microsoft Windows.
- Χωρίς την επέμβαση η διπλανή οθόνη αναφέρει για τη μνήμη 3069MB.



# Υποστήριξη PAE σε FreeBSD και Linux (kernel recompilation)

```
#
# PAE -- Generic kernel configuration file for FreeBSD/i386 PAE
#
# $FreeBSD: src/sys/i386/conf/PAE,v 1.32.2.1 2009/08/03 08:13:06 kensmith Exp $

include GENERIC

ident          PAE-GENERIC

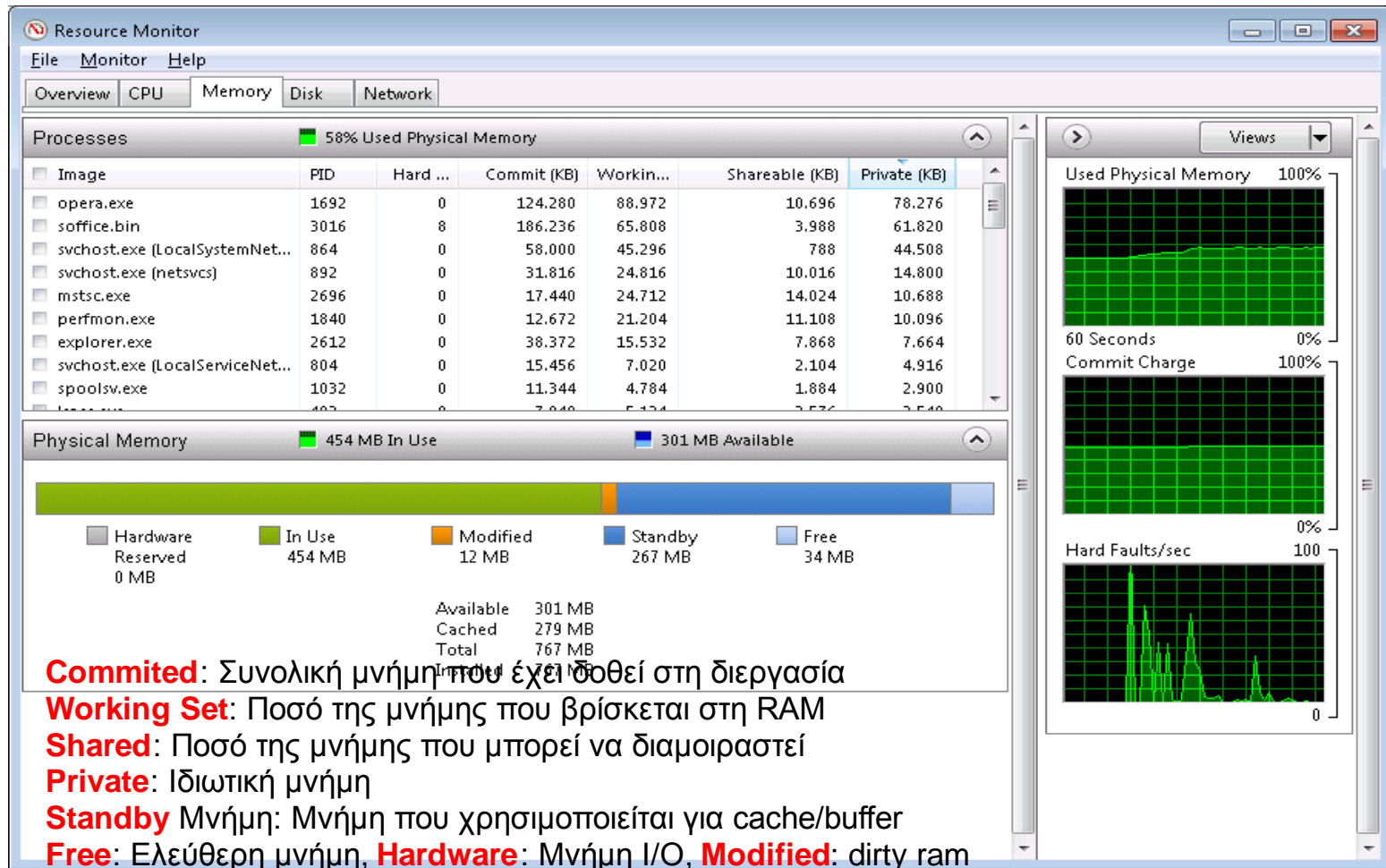
# To make a PAE kernel, the next option is needed
options        PAE                # Physical Address Extensions Kernel

# Compile acpi in statically since the module isn't built properly.  Most
# machines which support large amounts of memory require acpi.
device         acpi

[ ] Machine Check Exception
< > Check for non-fatal errors on AMD Athlon/Duron / Intel Pentium 4
[ ] Check for P4 thermal throttling interrupt.
< > Toshiba Laptop support
< > Dell laptop support
[ ] Enable X86 board specific fixups for reboot
< > /dev/cpu/microcode - Intel IA32 CPU microcode support
< > /dev/cpu/*/msr - Model-specific register support
< > /dev/cpu/*/cpuid - CPU information support
- High Memory Support (64GB) --->
-- PAE (Physical Address Extension) Support
   Memory model (Flat Memory) --->
-- 64 bit Memory and IO resources (EXPERIMENTAL)
[*] Allocate 3rd-level pagetables from highmem
[ ] Math emulation
[*] MTRR (Memory Type Range Register) support
[ ] EFI runtime service support
[ ] Enable kernel irq balancing
[*] Enable seccomp to safely compute untrusted bytecode
   Timer frequency (1000 HZ) --->
[*] kexec system call
```



# Καταστάσεις της μνήμης στα Windows 7



# Οι καταστάσεις της μνήμης στα Unix like

```
last pid: 50572; load averages: 0.16, 0.14, 0.09
116 processes: 1 running, 114 sleeping, 1 stopped
CPU: total memory, recently free, cannot paged-out, used in case, used in IO buffers, not used at all
Mem: 102M Active, 36M Inact, 3623M Wired, 94M Cache, 418M Buf, 88M Free
Swap: 4028M Total, 420M Used, 3608M Free, 10% Inuse, 264K Out
```

					committed	working set					
PID	USERNAME	THR	PRI	NICE	SIZE	RES	STATE	C	TIME	WCPU	COMMAND
55513	bigbrother	23	44	0	1034M	810M	ucond	0	20.2H	13.96%	VBoxHeadle
20281	bigbrother	1	44	0	19304K	2132K	select	1	11:19	0.00%	Xvnc
95237	mysql	10	76	0	51676K	2208K	sigwai	0	5:12	0.00%	mysqld
84189	nobody	1	44	0	9936K	1280K	select	1	3:19	0.00%	openvpn
56070	bigbrother	11	44	0	42396K	2916K	ucond	0	2:41	0.00%	VBoxSVC
1027	root	1	44	0	13200K	548K	select	0	2:25	0.00%	natd
25659	root	1	64	20	81664K	3372K	select	0	1:43	0.00%	perl5.8.9
33169	bigbrother	1	44	0	10528K	1352K	select	0	1:36	0.00%	screen
49365	bind	5	44	0	72564K	16556K	kqread	0	1:20	0.00%	named
35057	clamav	3	44	0	168M	18924K	ucond	1	1:15	0.00%	clamd
28270	www	1	44	0	12284K	0K	kqread	0	1:14	0.00%	<nginx>
18732	spamd	1	66	20	119M	52632K	select	1	0:51	0.00%	perl5.8.9
37972	root	1	44	r0	7016K	356K	nanslp	1	0:26	0.00%	watchdogd
89377	nobody	2	44	0	10016K	516K	select	1	0:25	0.00%	3proxy
55660	bigbrother	1	44	0	26192K	812K	select	0	0:23	0.00%	VBoxXPCOMI
13293	root	1	44	0	17000K	1788K	select	1	0:22	0.00%	mc
25133	bigbrother	1	44	0	37452K	1020K	select	0	0:19	0.00%	wmaker
34976	root	1	44	0	26300K	2872K	select	1	0:18	0.00%	smbd
34736	mailnull	4	55	0	16044K	1608K	sigwai	1	0:13	0.00%	milter-gre
64051	root	1	44	0	11068K	1048K	select	1	0:10	0.00%	sendmail
51526	root	1	54	10	5992K	668K	select	1	0:08	0.00%	syslogd
44843	bigbrother	1	44	0	22992K	936K	STOP	0	0:03	0.00%	mutt
30644	clayton	2	76	0	12612K	964K	sigwai	1	0:02	0.00%	clayton_mil





# FreeBSD Memory meter

```

Mem:KB      REAL          VIRTUAL          VN PAGER      SWAP PAGER
      Tot    Share    Tot    Share    Free
Act  130880    7452  2542504  44092  132324  count
All  2129688  27912 1076456k 109188    pages
Proc:
  r  p  d  s  w  Csw  Trp  Sys  Int  Sof  Flt      cow  Interrupts
      99 15 3696 109 40k  64 661      zfod  4070 total
1.9%Sys  0.0%Intr  1.5%User  0.0%Nice 86.6%Idle  %zfod  ata0 irq1
=====>      daefr  uhci2 18
                prcfr  uhci0 23
                dtbuf  20
                totfr  235  cpu0: tim
Namei      Name-cache  Dir-cache  143001 desvn  react  21 re0 irq25
  Calls    hits  %    hits  %    52031 numvn  pdwak  2003 cpu1: tim
    1      1 100      1 100    35750 frevn  pdpgs
Disks      ad0    da0    md0    md1    md2  pass0  3755500 wire
KB/t      122  0.00  0.00  0.00  0.00  0.00  23996 act
tps       37    0    0    0    0    0  125512  inact
MB/s     4.39  0.00  0.00  0.00  0.00  0.00  89756  cache
sbusy     8    0    0    0    0    0  42568  free
                428464  buf
  
```



# Παράδειγμα χρήσης μνήμης στο 64bit FreeBSD 8.2

```
SYSTEM MEMORY INFORMATION:
mem_wire:      3770621952 ( 3595MB) [ 91%] Wired: disabled for paging out
mem_active:    +   75923456 (   72MB) [  1%] Active: recently referenced
mem_inactive:  +   79593472 (   75MB) [  1%] Inactive: recently not referenced
mem_cache:     +  110395392 (  105MB) [  2%] Cached: almost avail. for allocation
mem_free:      +   97685504 (   93MB) [  2%] Free: fully available for allocation
mem_gap_vm:    +   1728512  (    1MB) [  0%] Memory gap: UNKNOWN
-----
mem_all:       =  4135948288 ( 3944MB) [100%] Total real memory managed
mem_gap_sys:   +  149995520 (  143MB)      Memory gap: Kernel?!
-----
mem_phys:     =  4285943808 ( 4087MB)      Total real memory available
mem_gap_hw:    +   9023488  (    8MB)      Memory gap: Segment Mappings?!
-----
mem_hw:       =  4294967296 ( 4096MB)      Total real memory installed

SYSTEM MEMORY SUMMARY:
mem_used:     4007292928 ( 3821MB) [ 93%] Logically used memory
mem_avail:    +  287674368 (  274MB) [  6%] Logically available memory
-----
mem_total:   =  4294967296 ( 4096MB) [100%] Logically total memory
```



# Τμηματοποίηση σταθερού μεγέθους (Fixed Partitioning) – ίσα τμήματα

Ίσου μεγέθους τμήματα (equal-size partitions):

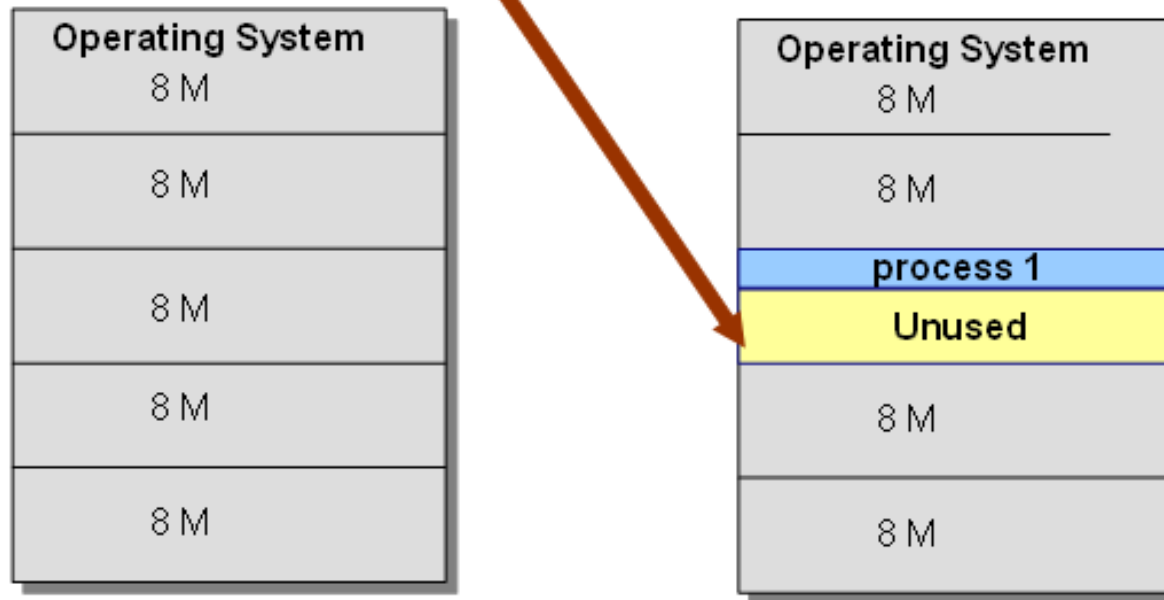
- Κάθε διεργασία με μέγεθος μικρότερο ή ίσο με το μέγεθος του τμήματος μπορεί να φορτωθεί στο διαθέσιμο τμήμα.
- Αν όλα τα τμήματα είναι γεμάτα, το Λ.Σ. μπορεί να κάνει εναλλαγή μιας διεργασίας.
- Ένα πρόγραμμα είναι πιθανό να μη χωρά σε ένα τμήμα. Ο προγραμματιστής πρέπει να σχεδιάσει το πρόγραμμα με επικαλύψεις.
- Με τη μέθοδο αυτή η χρήση της κύριας μνήμης είναι **εξαιρετικά αναποτελεσματική** (inefficient). Κάθε πρόγραμμα, όσο μικρό και να είναι, καταλαμβάνει ένα ολόκληρο τμήμα.
- Ο ανεκμετάλλετος χώρος εσωτερικά σε ένα τμήμα αναφέρεται ως **εσωτερικός κατακερματισμός** (internal fragmentation).





# Τα σταθερού μεγέθους τμήματα μνήμης οδηγούν στον εσωτερικό κατακερματισμό

- **Εσωτερικός κατακερματισμός** – μέρος του τμήματος που δεν χρησιμοποιείται.



# Πλεονεκτήματα και μειονεκτήματα της χρήσης σταθερού μεγέθους τμημάτων μνήμης

---

## Πλεονεκτήματα:

- Μικρή επιβάρυνση στο Λ.Σ.

## Μειονεκτήματα:

- Ανεπαρκής χρήση της μνήμης λόγω του εσωτερικού κατακερματισμού που κρατά σταθερό τον μέγιστο αριθμό διεργασιών που μπορούν να εκτελεστούν.
- Οι μικρές διεργασίες δεν χρησιμοποιούν αποτελεσματικά τον χώρο των τμημάτων.

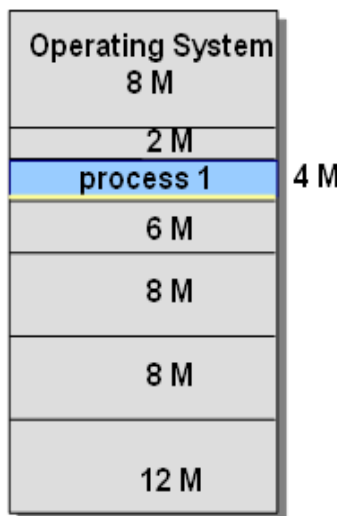


# Τμηματοποίηση σταθερού μεγέθους – με πολλαπλά άνισα τμήματα

Μειώνει τα προβλήματα της τμηματοποίησης ίσων τμημάτων.

## Πλεονεκτήματα

Δεν υπάρχει εσωτερικός κατακερματισμός. Περισσότερο αποτελεσματική χρήση της κύριας μνήμης, σε σχέση με τη μέθοδο των ίσων τμημάτων.



## Μειονεκτήματα

Ανεπαρκής χρήση του επεξεργαστή λόγω της ανάγκης για συμπίεση για την αντιμετώπιση του **εξωτερικού κατακερματισμού** (τμήματα της μνήμης που δε χρησιμοποιούνται λόγω μεγέθους).



# Εσωτερικός και Εξωτερικός Κατακερματισμός

---

- **ΕΣΩΤΕΡΙΚΟΣ:** είναι η μνήμη που δεν χρησιμοποιείται (δαπανάται) και είναι ορατή μόνον από τη διεργασία που ζητά μνήμη. Συμβαίνει επειδή η ποσότητα μνήμης που θα εκχωρηθεί στη διεργασία πρέπει να είναι μεγαλύτερη ή ίση από την αιτούμενη ποσότητα.
- **ΕΞΩΤΕΡΙΚΟΣ:** είναι η μνήμη που δεν χρησιμοποιείται (δαπανάται) και είναι ορατή από το σύστημα εκτός των διεργασιών που απαιτούν μνήμη. Συμβαίνει επειδή όλες οι απαιτήσεις μνήμης δεν είναι του ίδιου μεγέθους.



# Αλγόριθμοι τοποθέτησης

---

## Ίσου μεγέθους τμήματα:

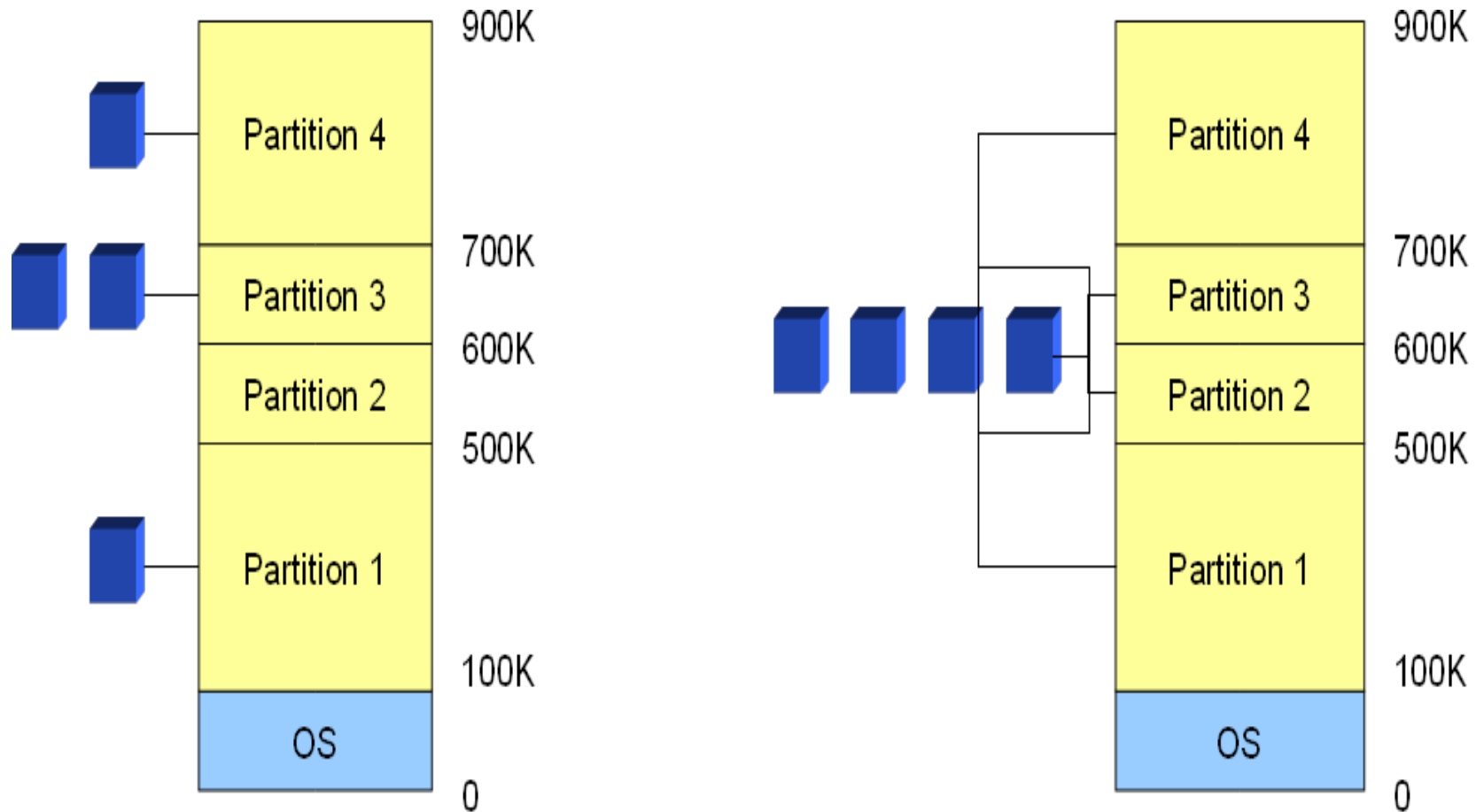
- Επειδή όλα τα τμήματα είναι ίσου μεγέθους, δεν έχει σημασία ποιο χρησιμοποιείται.
- Αν είναι όλα κατειλημμένα γίνεται εναλλαγή (swapping).

## Διαφορετικού μεγέθους τμήματα:

- **Ουρά για κάθε τμήμα:**
  - Κάθε διεργασία μπορεί να αντιστοιχηθεί στο μικρότερο τμήμα στο οποίο χωρά.
  - Οι διεργασίες αντιστοιχούνται με τρόπο ώστε να ελαχιστοποιείται η σπατάλη μνήμης μέσα σε ένα τμήμα (μείωση του εσωτερικού κατακερματισμού).
- **Μια μοναδική ουρά για όλες τις διεργασίες:**
  - Όταν η διεργασία πρέπει να φορτωθεί στη μνήμη επιλέγεται το μικρότερο διαθέσιμο τμήμα.
  - καλύτερη ικανότητα για τη βελτιστοποίηση χρήσης της CPU.



# Χρήση μοναδικής ή πολλαπλής ουράς



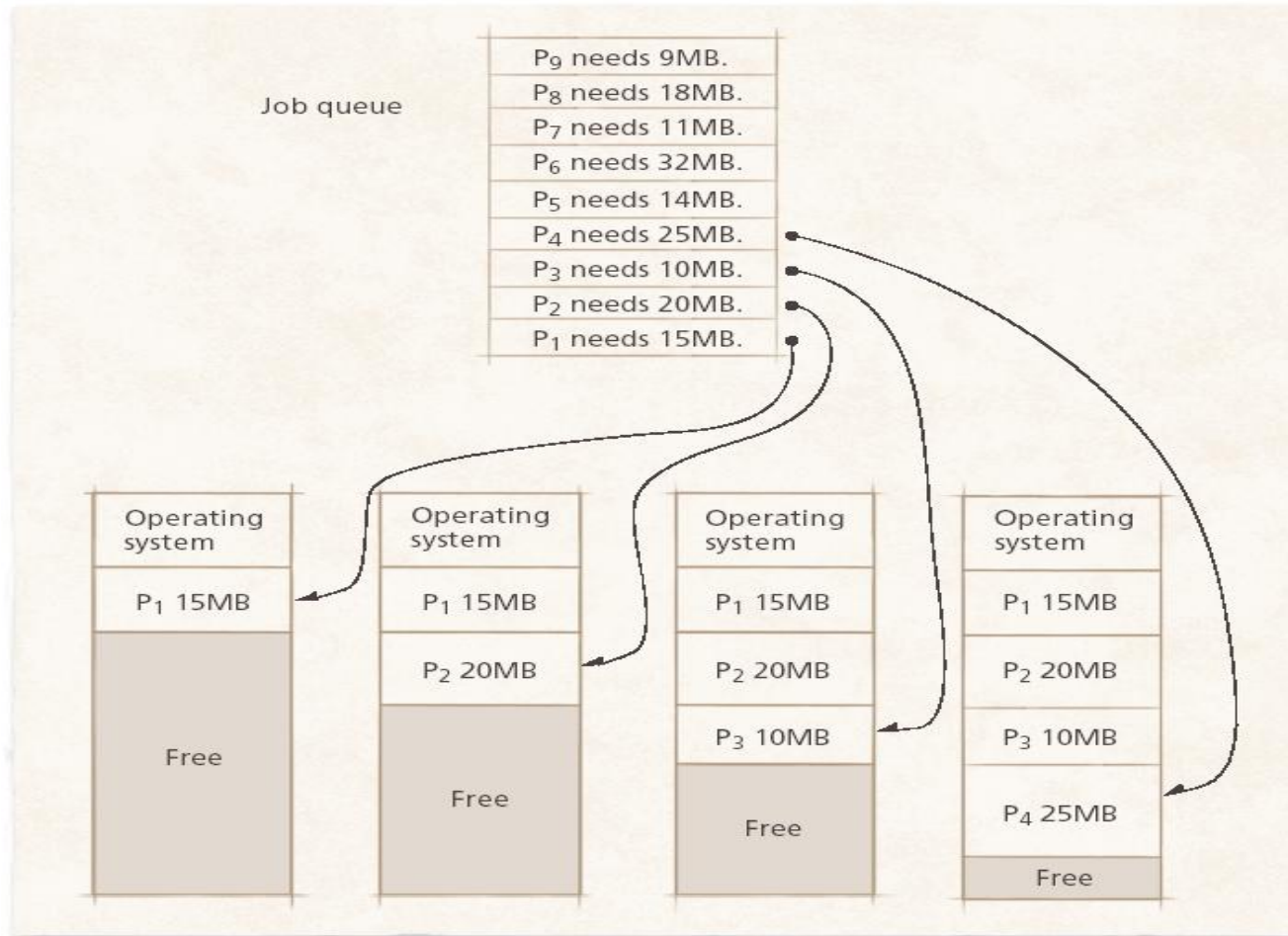
# Δυναμική τμηματοποίηση

---

- Τμήματα μεταβλητού μεγέθους και πλήθους.
- Μια διεργασία αντιστοιχείται ακριβώς στην ποσότητα μνήμης που απαιτείται.
- Τελικά υπάρχουν κενά στη μνήμη. Αυτός είναι ο **εξωτερικός κατακερματισμός (external fragmentation)**.
- Πρέπει να χρησιμοποιηθεί **συμπύεση (compaction)** που θα μετατοπίσει τις διεργασίες έτσι ώστε να είναι συνεχόμενες και όλη η ελεύθερη μνήμη να αποτελεί μια ενότητα (block).
- Η συμπύεση σπαταλά το χρόνο της CPU και προϋποθέτει τη δυνατότητα δυναμικής μετατόπισης (μεταφορά ενός προγράμματος σε άλλη περιοχή μνήμης χωρίς να ακυρώνονται οι αναφορές της μνήμης).

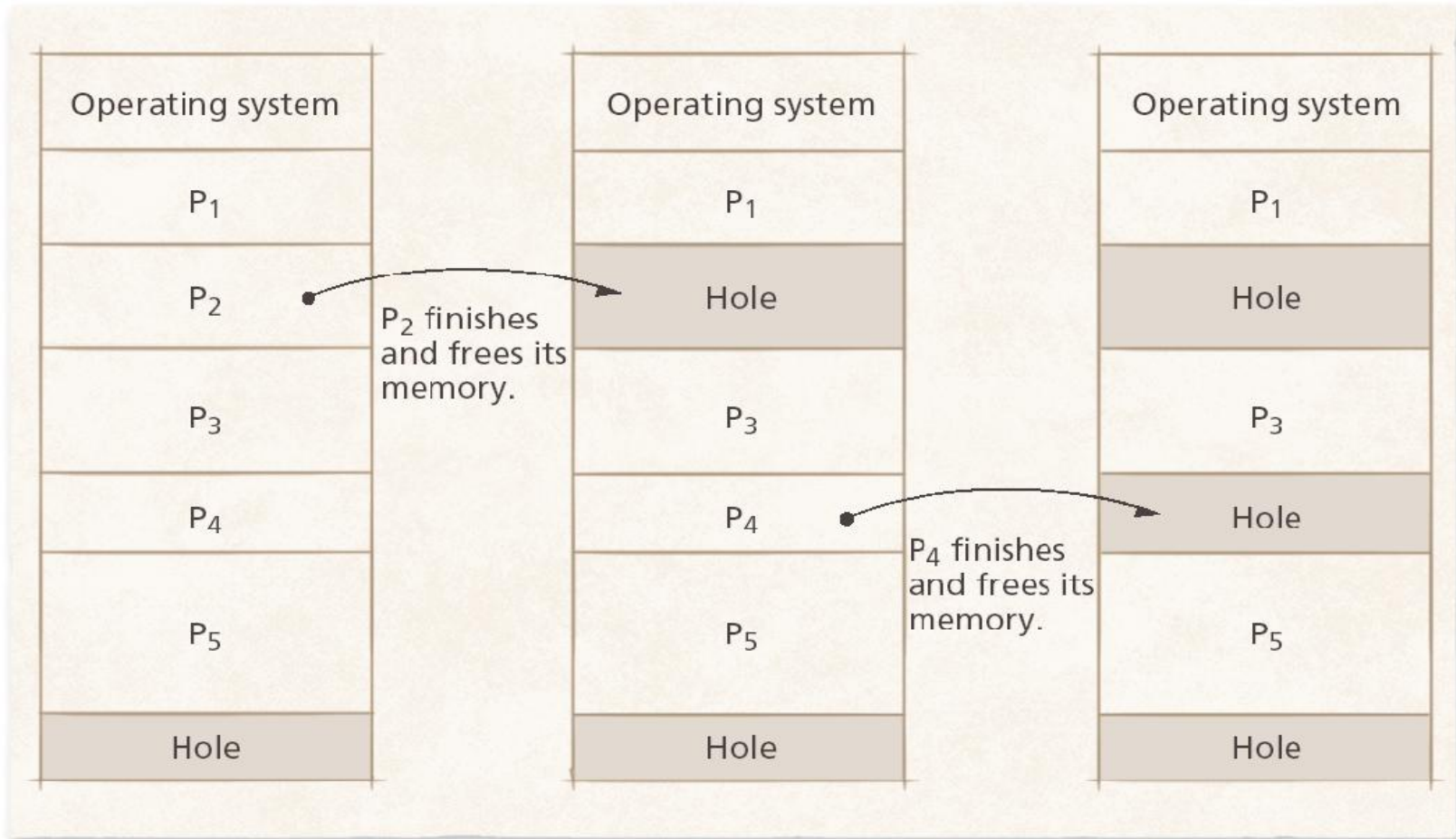


# Παράδειγμα δυναμικής τμηματοποίησης (1/2)





# Παράδειγμα δυναμικής τμηματοποίησης (2/2)



# Αλγόριθμοι τοποθέτησης δυναμικής τμηματοποίησης (1/4)

---

Το Λ.Σ. πρέπει να αποφασίσει ποιο ελεύθερο τμήμα της μνήμης θα εκχωρήσει σε μια διεργασία.

Οι κυριότεροι είναι:

- 1) **Αλγόριθμος καλύτερης τοποθέτησης** (best-fit algorithm).
  - Επιλογή του block που είναι **πλησιέστερα** στο μέγεθος που απαιτείται.
  - Έχει τη **χειρότερη** απόδοση.
  - Μια και βρίσκει το μικρότερο block για τη διεργασία, η κύρια μνήμη γεμίζει γρήγορα από blocks που είναι πολύ μικρά και η συμπίεση πραγματοποιείται πιο συχνά.



# Αλγόριθμοι τοποθέτησης δυναμικής τμηματοποίησης (2/4)

---

## 2) Αλγόριθμος πρώτης τοποθέτησης (First-fit algorithm).

- Ξεκινά και σαρώνει τη μνήμη από την αρχή και επιλέγει το πρώτο διαθέσιμο μπλοκ που είναι αρκετά μεγάλο.
- Ταχύτερος.
- Πολλές διεργασίες φορτώνονται στο εμπρός τμήμα της μνήμης που θα πρέπει να εξετάζεται κάθε φορά που γίνεται προσπάθεια για την εύρεση ενός ελεύθερου block.



# Αλγόριθμοι τοποθέτησης δυναμικής τμηματοποίησης (3/4)

## 3) Αλγόριθμος επόμενης τοποθέτησης (next-fit algorithm).

- Ξεκινά να σαρώνει τη μνήμη από την τελευταία τοποθέτηση και επιλέγει το επόμενο αρκετά μεγάλο διαθέσιμο μπλοκ.
- Εκχωρεί συχνά ένα block μνήμης που βρίσκεται στο τέλος της μνήμης, όπου βρίσκεται το μεγαλύτερο block.
- Το μεγαλύτερο block μνήμης διασπάται σε μικρότερα blocks.
- Η συμπίεση απαιτείται για να αποκτηθεί ένα μεγάλο block στο τέλος της μνήμης.



# Αλγόριθμοι τοποθέτησης δυναμικής τμηματοποίησης (4/4)

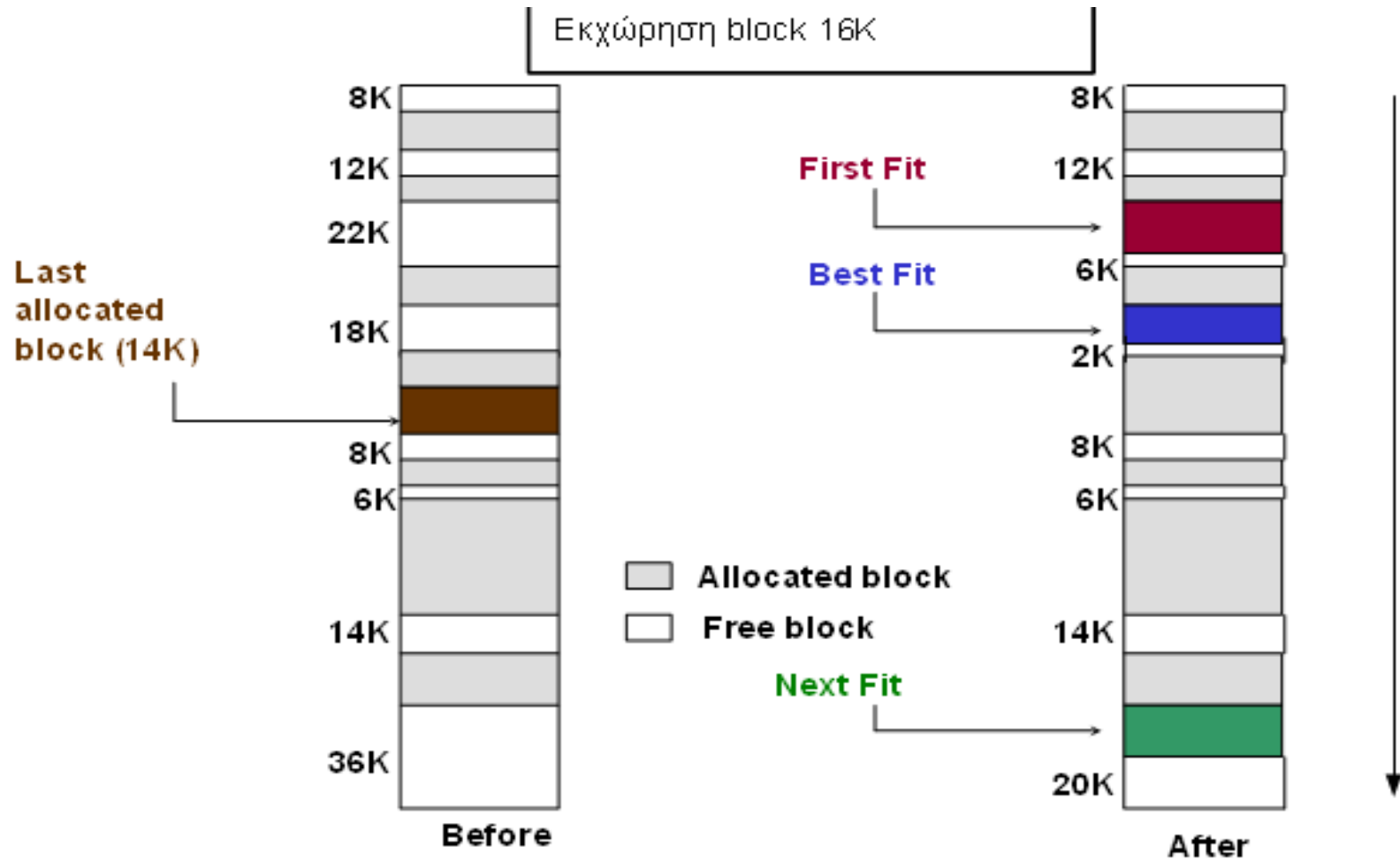
---

## 4) Αλγόριθμος χειρότερης τοποθέτησης (worst-fit algorithm).

- Επιλέγει το μέγιστο διαθέσιμο μπλοκ που υπάρχει.
- Αναζήτηση σε όλη τη μνήμη.
- Καταλήγει στο μέγιστο δυνατό υπόλοιπο ανοιγμάτων.



# Παράδειγμα των αλγορίθμων τοποθέτησης



# Σύστημα ζευγών ή φίλων (buddies)

---

- Όλος ο διαθέσιμος χώρος της μνήμης συμπεριφέρεται ως ένα μοναδικό block μεγέθους  $2^U$ .
- Αν απαιτηθεί ένα block μεγέθους  $s$  τέτοιου ώστε  $2^{U-1} < s \leq 2^U$ , εκχωρείται ολόκληρο.
- Διαφορετικά το block διαιρείται σε δύο ίσα τμήματα (ζεύγος φίλων – buddies).
- Η διαδικασία συνεχίζεται μέχρι να δημιουργηθεί το μικρότερο block που είναι μεγαλύτερο ή ίσο του μεγέθους  $s$ .



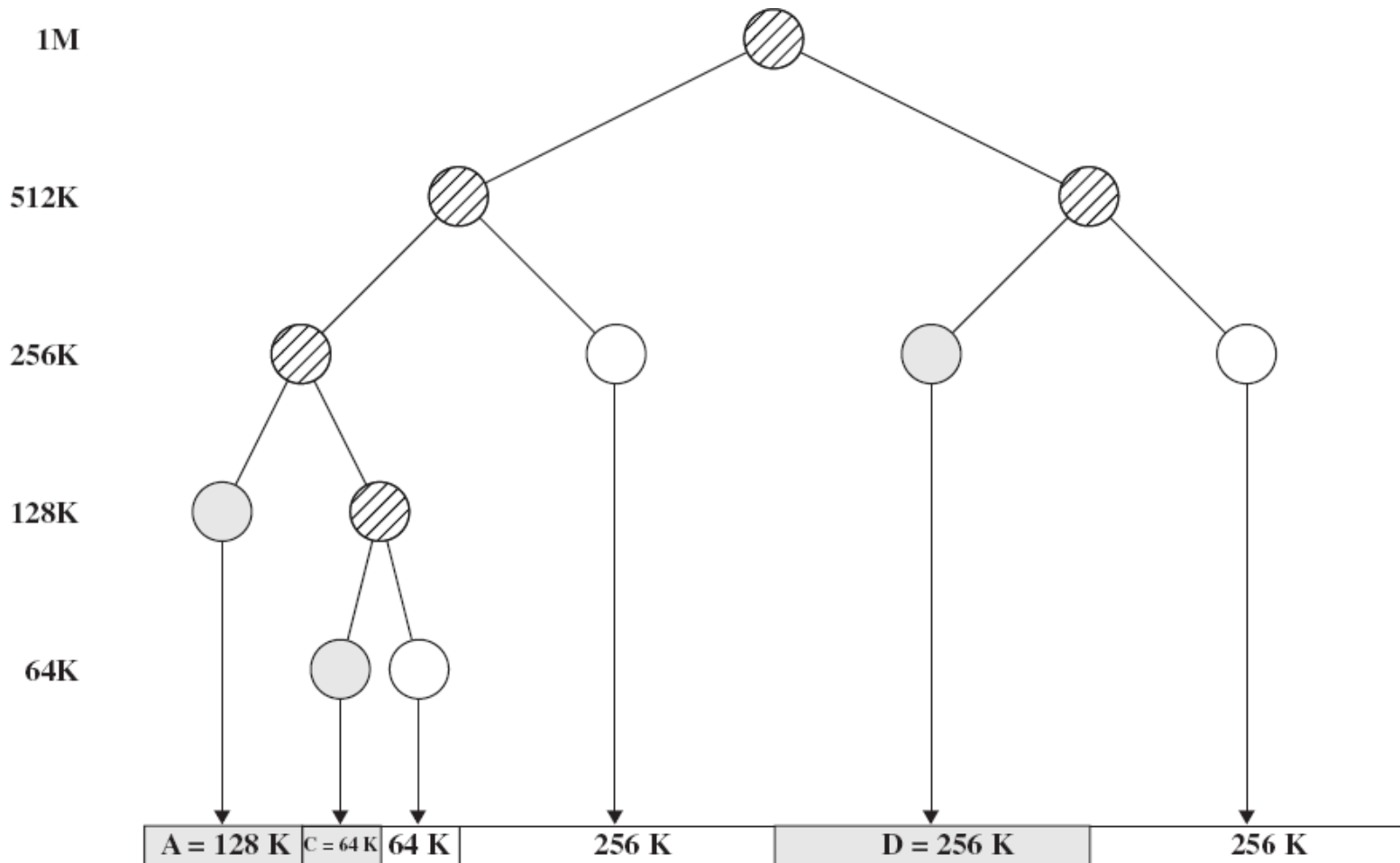
# Παράδειγμα Σύστημα ζευγών

1 Mbyte block	1 M					
Request 100 K	A = 128 K	128 K	256 K	512 K		
Request 240 K	A = 128 K	128 K	B = 256 K	512 K		
Request 64 K	A = 128 K	C = 64 K	64 K	B = 256 K	512 K	
Request 256 K	A = 128 K	C = 64 K	64 K	B = 256 K	D = 256 K	256 K
Release B	A = 128 K	C = 64 K	64 K	256 K	D = 256 K	256 K
Release A	128 K	C = 64 K	64 K	256 K	D = 256 K	256 K
Request 75 K	E = 128 K	C = 64 K	64 K	256 K	D = 256 K	256 K
Release C	E = 128 K	128 K	256 K	D = 256 K	256 K	
Release E	512 K				D = 256 K	256 K
Release D	1 M					





# Δενδρική αναπαράσταση του συστήματος ζευγών



---

# Swapping



# Εναλλαγή

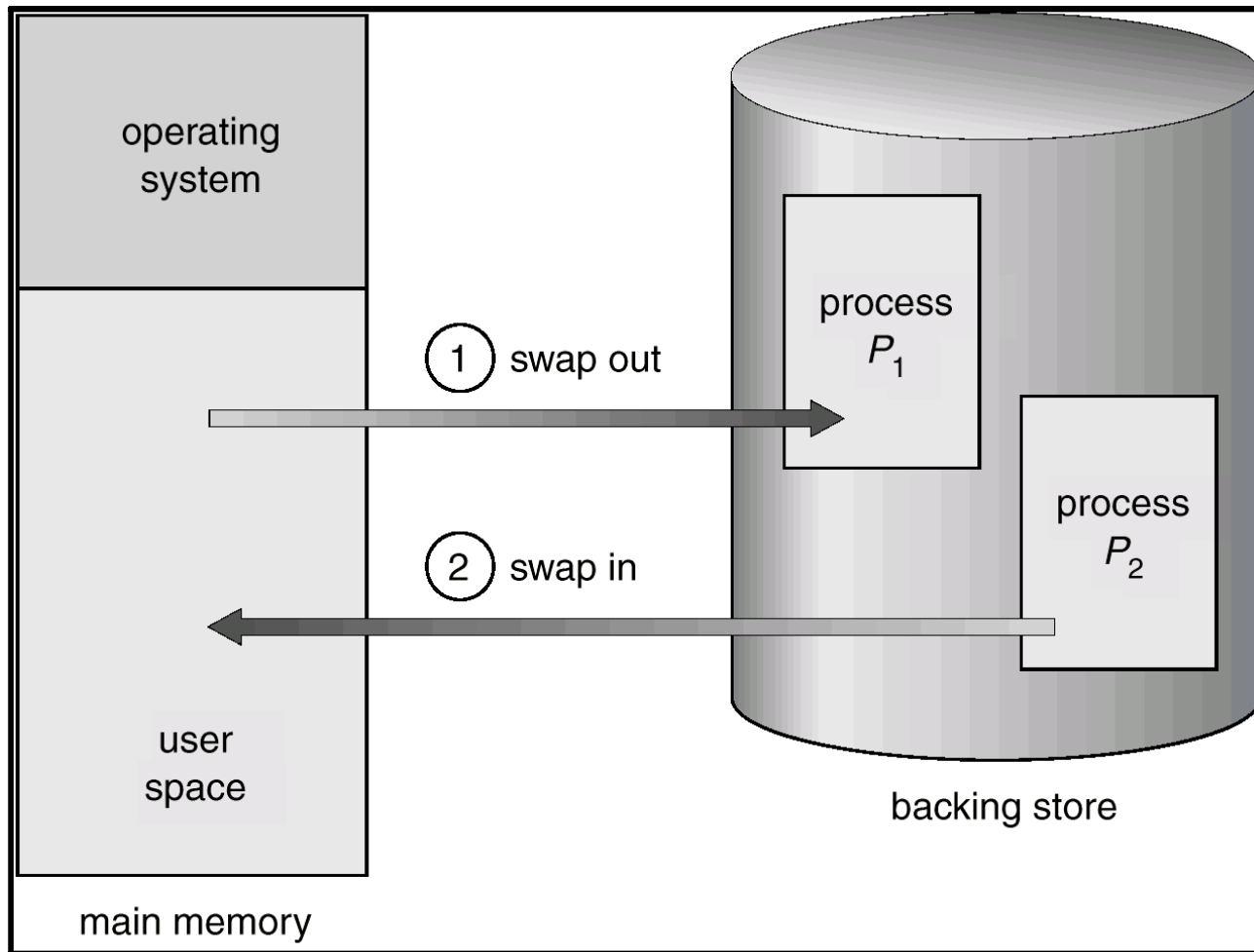
## ή Ανταλλαγή ή Swapping (1/3)

---

- Μία διεργασία μπορεί να μεταφερθεί προσωρινά από τη μνήμη σε κάποιο υποστηρικτικό μέσο αποθήκευσης και στη συνέχεια να φορτωθεί ξανά στη μνήμη, ώστε να συνεχίσει την εκτέλεση της.
- Υποστηρικτικό μέσο αποθήκευσης (backing store) : γρήγορος δίσκος μεγάλης χωρητικότητας ώστε να χωράει όλα τα αντίγραφα των εικόνων όλων των χρηστών.
- Πρέπει να παρέχει απευθείας πρόσβαση σε αυτές τις εικόνες της μνήμης.
- Κύλιση προς τα έξω, Κύλιση προς τα μέσα : παράμετρος ανταλλαγής που χρησιμοποιείται για αλγόριθμους δρομολόγησης που βασίζονται σε προτεραιότητες. **Η διεργασία χαμηλότερης προτεραιότητας μετακινείται εκτός, ώστε να φορτωθεί και να εκτελεσθεί η διεργασία υψηλότερης προτεραιότητας.**
- Ο κυριότερος παράγοντας του χρόνου ανταλλαγής είναι ο **χρόνος μεταφοράς** (transfer time). Ο συνολικός χρόνος μεταφοράς είναι ανάλογος του ποσού της μνήμης που μετακινείται και ανταλλάσσεται.
- Τροποποιημένες εκδοχές της ανταλλαγής υπάρχουν σε πολλά συστήματα π.χ. UNIX, Microsoft Windows.



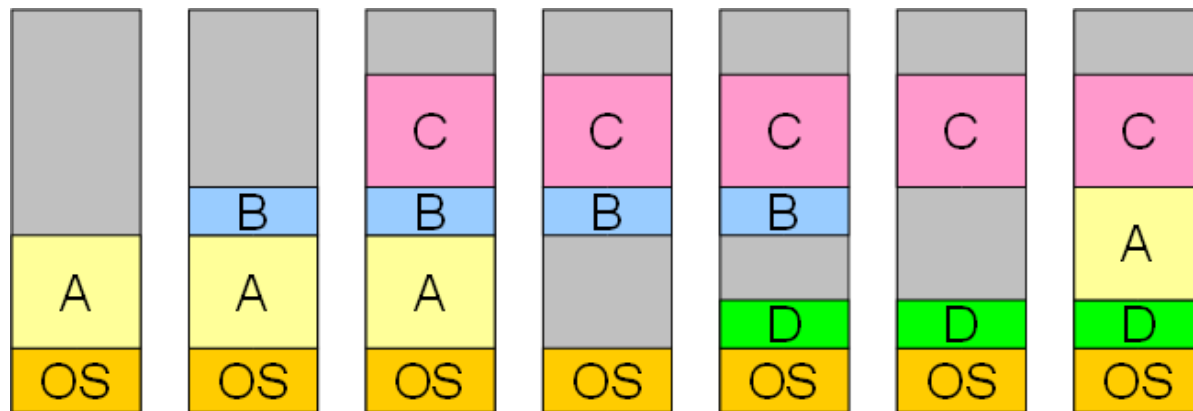
# Εναλλαγή ή Ανταλλαγή ή Swapping (2/3)



# Εναλλαγή ή Ανταλλαγή ή Swapping (3/3)

Η εκχώρηση μνήμης αλλάζει καθώς:

- Έρχονται νέες διεργασίες στη μνήμη.
- Διεργασίες εγκαταλείπουν τη μνήμη.
- Εναλλάσσονται στο δίσκο.
- Ολοκληρώνουν την εκτέλεσή τους.
- Οι γκρι περιοχές είναι μνήμη που δεν χρησιμοποιείται.



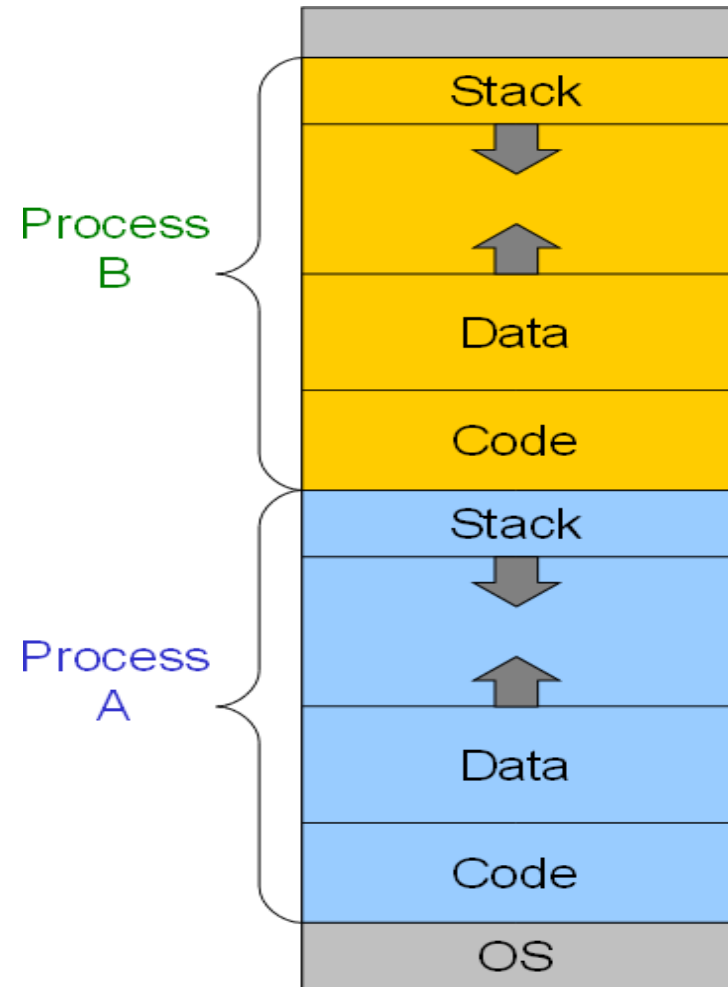
# Εναλλαγή: αφήνοντας χώρο για ανάπτυξη

Ανάγκη για τη δυνατότητα ανάπτυξης των προγραμμάτων.

- Εκχώρηση περισσότερης μνήμης για δεδομένα.
- Μεγαλύτερο stack.

Εκχώρηση μεγαλύτερης ποσότητας μνήμης από όση απαιτείται καταρχήν.

- Αναποτελεσματική: σπατάλη μνήμης που δεν χρησιμοποιείται.
- Τι θα συμβεί αν η διεργασία απαιτεί υπερβολική ποσότητα μνήμης;



# Περιορισμοί της εναλλαγής

---

- **Προβλήματα** με την εναλλαγή.
  - Η διεργασία πρέπει να χωρά στη φυσική μνήμη (αδύνατη η εκτέλεση μεγαλύτερων διεργασιών).
  - Η μνήμη κατακερματίζεται (fragmented).
    - Εξωτερικός κατακερματισμός: πλήθος μικρών ελεύθερων περιοχών.
    - Απαιτείται συμπίεση για την επανασυναρμολόγηση μεγαλύτερων ελεύθερων περιοχών.
- Οι διεργασίες μπορούν να βρίσκονται και στη μνήμη και στο δίσκο.
- Οι **επικαλύψεις επιλύουν** το πρώτο πρόβλημα.
  - Κατανέμουν τη διεργασία στη διάρκεια του χρόνου (κυρίως τα δεδομένα).
  - Δεν επιλύουν το πρόβλημα του κατακερματισμού.



# Άσκηση (1/4)

---

Υποθέστε ότι έχετε ελεύθερη μνήμη σε τμήματα μεγέθους 100KB, 500KB, 200KB, 300KB, and 600KB (με αυτή τη σειρά) και υπάρχουν κατά σειρά απαιτήσεις μνήμης για 212KB, 417KB, 112KB, και 426KB.

- Δείξτε πως ο αλγόριθμος πρώτης τοποθέτησης θα διευθετήσει τις απαιτήσεις στην ελεύθερη μνήμη. Υπολογίστε το ελεύθερο τμήμα μνήμης μετά από κάθε άφιξη.
- Δείξτε πως ο αλγόριθμος βέλτιστης τοποθέτησης θα διευθετήσει τις ίδιες απαιτήσεις στην ελεύθερη μνήμη. Υπολογίστε το ελεύθερο τμήμα μνήμης μετά από κάθε άφιξη.
- Δείξτε πως ο αλγόριθμος επόμενης τοποθέτησης θα διευθετήσει τις απαιτήσεις στην ελεύθερη μνήμη (last allocated block 200KB). Υπολογίστε το ελεύθερο τμήμα μνήμης μετά από κάθε άφιξη.





# Άσκηση (2/4)

- Ένα σύστημα τοποθετεί διεργασίες στη μνήμη χρησιμοποιώντας δυναμική πολιτική τοποθέτησης. Κατά την πλέον πρόσφατη χρονική στιγμή έγινε φόρτωση μιας διεργασίας που χρειαζόταν 12KB μνήμης και η εικόνα μνήμης του συστήματος διαμορφώθηκε ως εξής:



- Οι σκιασμένες περιοχές δηλώνουν χρησιμοποιημένα τμήματα μνήμης, οι λευκές τα κενά τμήματα ενώ η περιοχή με μαύρο χρώμα τη θέση όπου έγινε η τελευταία τοποθέτηση. Οι αριθμοί δηλώνουν το μέγεθος σε KB.
- => Να σχεδιάσετε την εικόνα μνήμης μετά την τοποθέτηση μιας νέας διεργασίας που χρειάζεται 22KB μνήμης σύμφωνα με τους αλγορίθμους : first-fit, best-fit, next-fit.



# Άσκηση (3/4)

---

Ο διαχειριστής μνήμης ενός συστήματος που χρησιμοποιεί στρατηγική τμημάτων μεταβλητού μεγέθους διαθέτει ελεύθερα τμήματα μεγέθους 600, 400, 1000, 2200, 1600, 1050 bytes. Θεωρείστε κατά σειρά τις παρακάτω διαδοχικές απαιτήσεις:

- ποιο τμήμα θα επιλεγεί για απαίτηση 1603 bytes με τη μέθοδο best-fit;
- ποιο τμήμα θα επιλεγεί για απαίτηση 949 bytes με τη μέθοδο best-fit;
- ποιο τμήμα θα επιλεγεί για απαίτηση 963 bytes με τη μέθοδο first-fit;
- ποιο τμήμα θα επιλεγεί για απαίτηση 349 bytes με τη μέθοδο first-fit;
- Υποθέστε ότι η λίστα ελεύθερων τμημάτων διατάσσεται κατά αύξουσα σειρά μεγέθους των τμημάτων που αναφέρονται στην αρχή της άσκησης. Ποιο τμήμα θα επιλεγεί για απαίτηση 1603 bytes με τη μέθοδο first-fit;



# Άσκηση (4/4)

---

- Θεωρείστε ένα σύστημα με 1MB διαθέσιμης μνήμης και απαιτήσεις για 42KB, 396KB, 10KB, και 28KB. Δείξτε το ποσό μνήμης που εκχωρείται για κάθε απαίτηση και την κατάσταση της μνήμης μετά από κάθε απαίτηση , **χρησιμοποιώντας το σύστημα ζευγών.**
- Πόσος εσωτερικός κατακερματισμός υπάρχει με το σενάριο αυτό;
- Πόσος εξωτερικός κατακερματισμός υπάρχει με το σενάριο αυτό;



---

# Τέλος Ενότητας



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

