

Κύρια Μνήμη: Διαχείριση, Τμηματοποίηση, Σελιδοποίηση, Κατάτμηση

Ανδρέας Λ. Συμεωνίδης

Αν. Καθηγητής

Τμήμα Ηλεκτρολόγων Μηχ/κών
&

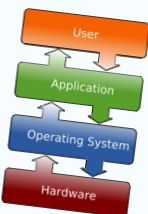
Μηχ/κών Υπολογιστών, Α.Π.Θ.

Email: asymeon@eng.auth.gr



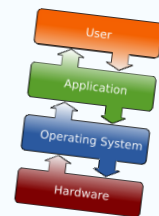
Μέρος Τρίτο

- Μνήμη
 - Διαχείριση μνήμης
 - Ιεραρχία μνήμης
 - Στρατηγικές διαχείρισης μνήμης
 - Μνήμη και πολυπρογραμματισμός
 - Τμηματοποίηση μνήμης
- Ιδεατή Μνήμη
 - Σύνδεση και Φόρτωση
 - Ιδεατές και πραγματικές διευθύνσεις
 - Λογική οργάνωση
 - Τμηματοποίηση ιδεατής μνήμης



Στόχοι της Δ-7

- Να ορίσει τις βασικές απαιτήσεις που πρέπει να ικανοποιεί η διαχείριση μνήμης
- Να παρουσιάσει τις βασικές στρατηγικές διαχείρισης μνήμης
- Να συζητήσει τις διάφορες τεχνικές τμηματοποίησης της μνήμης
- Να παρουσιάσει το σύστημα ζευγών φίλων
- Να ορίσει τις έννοιες της σελίδας και του πλαισίου
- Να ξεχωρίσει ανάμεσα στη Λογική, τη Σχετική και τη Φυσική Διεύθυνση



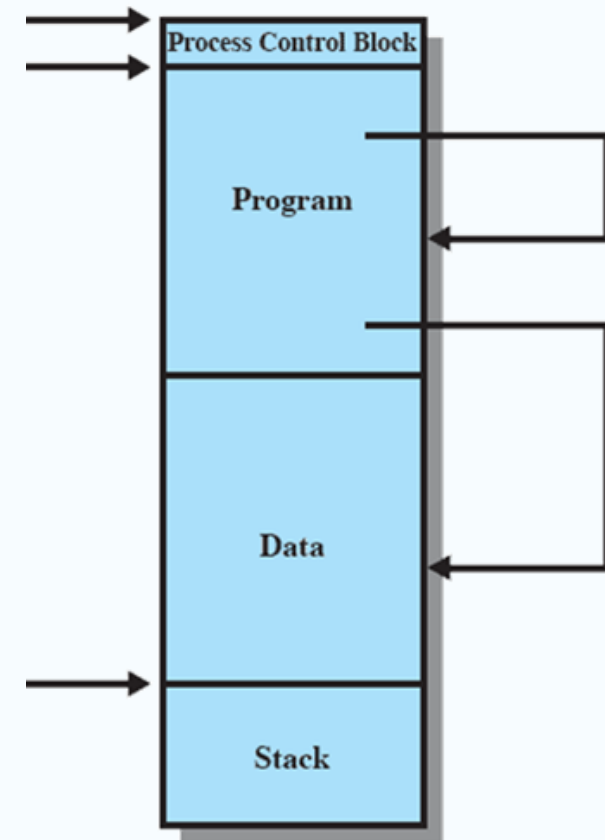
Εισαγωγή

- 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.
- 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 Διαχείρισης Μνήμης

- Μετατόπιση (Relocation)
 - Ο προγραμματιστής δε γνωρίζει πού στη μνήμη θα τοποθετηθεί, όταν αυτό θα εκτελείται
 - Όσο το πρόγραμμα εκτελείται, μπορεί να εναλλαγεί ανάμεσα στο δίσκο και στην κύρια μνήμη, και να επιστρέψει σε άλλη τοποθεσία
 - Οι αναφορές στη μνήμη πρέπει να μεταφράζονται σε κώδικα σε πραγματική φυσική διεύθυνση μνήμης



Απαιτήσεις Διαχείρισης Μνήμης (συν.)

- Προστασία
 - Οι διεργασίες δεν πρέπει να μπορούν να κάνουν αναφορά σε θέσεις μνήμης άλλης διεργασίας χωρίς άδεια
 - Είναι αδύνατον να γίνεται έλεγχος της απόλυτης διεύθυνσης κατά τη μεταγλώττιση – πρέπει να γίνεται έλεγχος κατά την εκτέλεση
- Διαμοίραση
 - Επιτρέπει πολλές διεργασίες να προσπελαύνουν το ίδιο τμήμα της μνήμης παράλληλα
 - Προτιμότερο να επιτρέπεται πρόσβαση κάθε διεργασίας στο ίδιο αντίγραφο του προγράμματος, παρά να έχει κάθε μια το δικό της αντίγραφο



Απαιτήσεις Διαχείρισης Μνήμης (συν.)

- Λογική Οργάνωση
 - Τα προγράμματα γράφονται σε ενότητες
 - Οι ενότητες μπορούν να γραφούν και να μεταγλωττιστούν ανεξάρτητα
 - Μπορούν να οριστούν διαφορετικοί βαθμοί ασφάλειας (read-only, execute-only)
 - Μπορεί να γίνει διαμοιρασμός τμημάτων ανάμεσα σε διεργασίες
- Φυσική Οργάνωση
 - Η μνήμη που είναι διαθέσιμη για ένα πρόγραμμα (+ δεδομένα) μπορεί να είναι ανεπαρκής
 - ◆ Η επικάλυψη επιτρέπει διαφορετικά τμήματα να ορίζονται στον ίδιο χώρο μνήμης
 - Ο προγραμματιστής δε γνωρίζει πόσος χώρος στη μνήμη θα είναι διαθέσιμος ανά πάσα στιγμή.

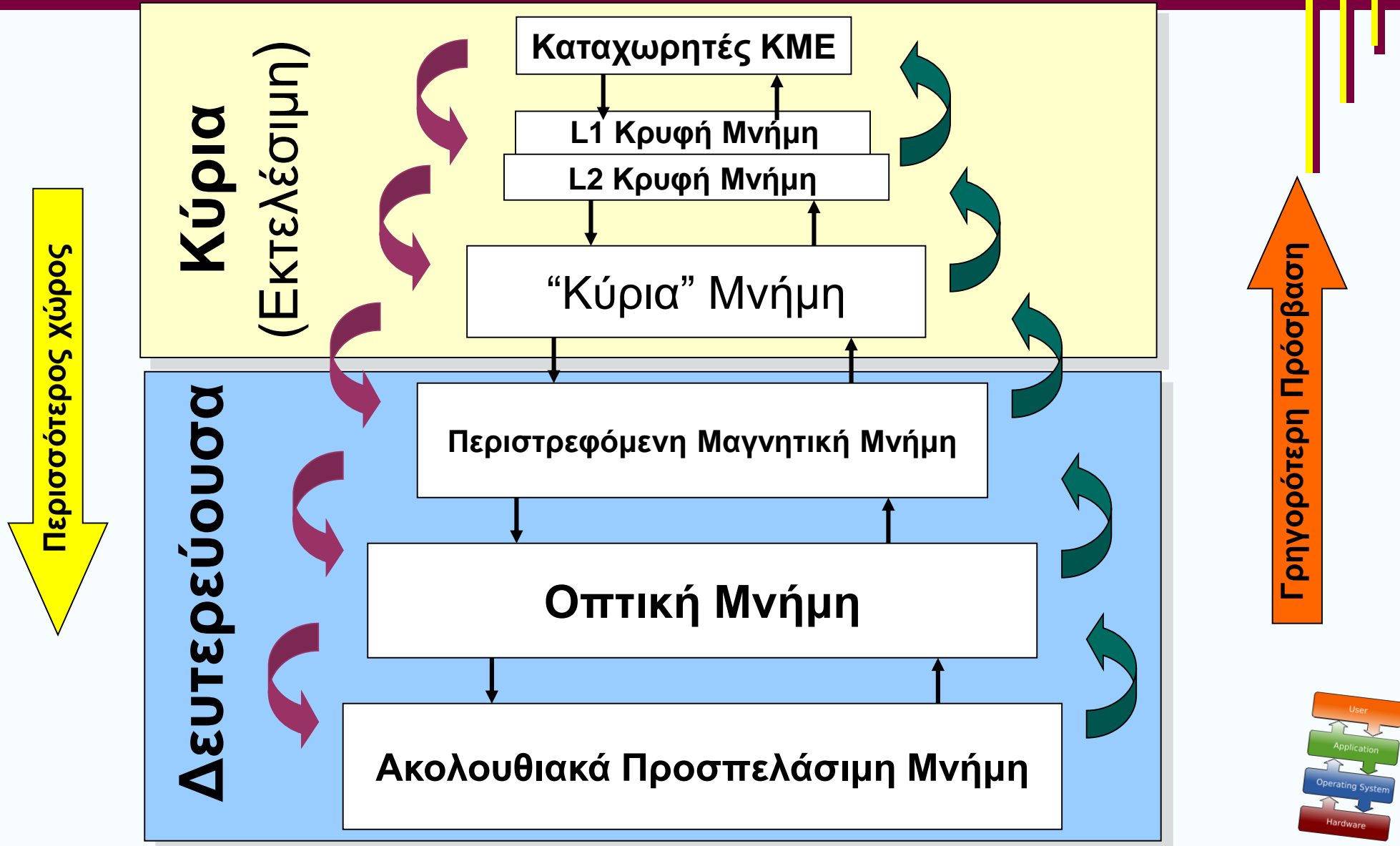


Διαχειριστής μνήμης

- Ο διαχειριστής μνήμης είναι ένα συστατικό του ΛΣ που ασχολείται με την οργάνωση και τις στρατηγικές διαχείρισης της μνήμης.
- 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.



Ιεραρχία Μνήμης



Στρατηγικές διαχείρισης μνήμης

- Σχεδιάζονται έτσι ώστε να είναι εφικτή η βέλτιστη δυνατή χρήση της κύριας μνήμης και διακρίνονται σε:

- **Στρατηγικές προσκόμισης**

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

- **Στρατηγικές τοποθέτησης**

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

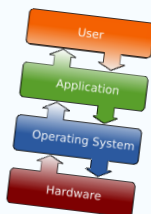
- **Στρατηγικές επανατοποθέτησης**

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.



Συνεχόμενη και μη συνεχόμενη εκχώρηση μνήμης

- Η συνεχόμενη εκχώρηση μνήμης δεν διαιρεί το πρόγραμμα
 - Αν το πρόγραμμα ήταν μεγαλύτερο από τη διαθέσιμη μνήμη το σύστημα δεν μπορούσε να το εκτελέσει
 - Αφορά προγενέστερα υπολογιστικά συστήματα
- Στη μη συνεχόμενη εκχώρηση μνήμης το πρόγραμμα διαιρείται σε τεμάχια ή τμήματα (code segments) που τοποθετούνται σε μη γειτονικές σχισμές στην κύρια μνήμη.
 - Η τεχνική αυτή κάνει εφικτή τη χρήση περιοχών που είναι πολύ μικρές για να χωρέσουν ολόκληρο πρόγραμμα.
 - Με τον τρόπο αυτό εισάγεται στο σύστημα πολυπλοκότητα



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 μέρος της κύριας μνήμης για να καλύψει μια ανάγκη (δηλ. να φορτώσει μια άλλη ενότητα)



Μονοπρογραμματισμός

- Τρεις απλοί τρόποι οργάνωσης μνήμης σε ΛΣ με μια διεργασία χρήστη
- Η προστασία μνήμης δεν τίθεται ως πρόβλημα

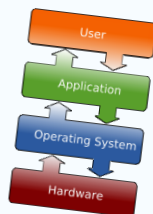
0xFFFF

0

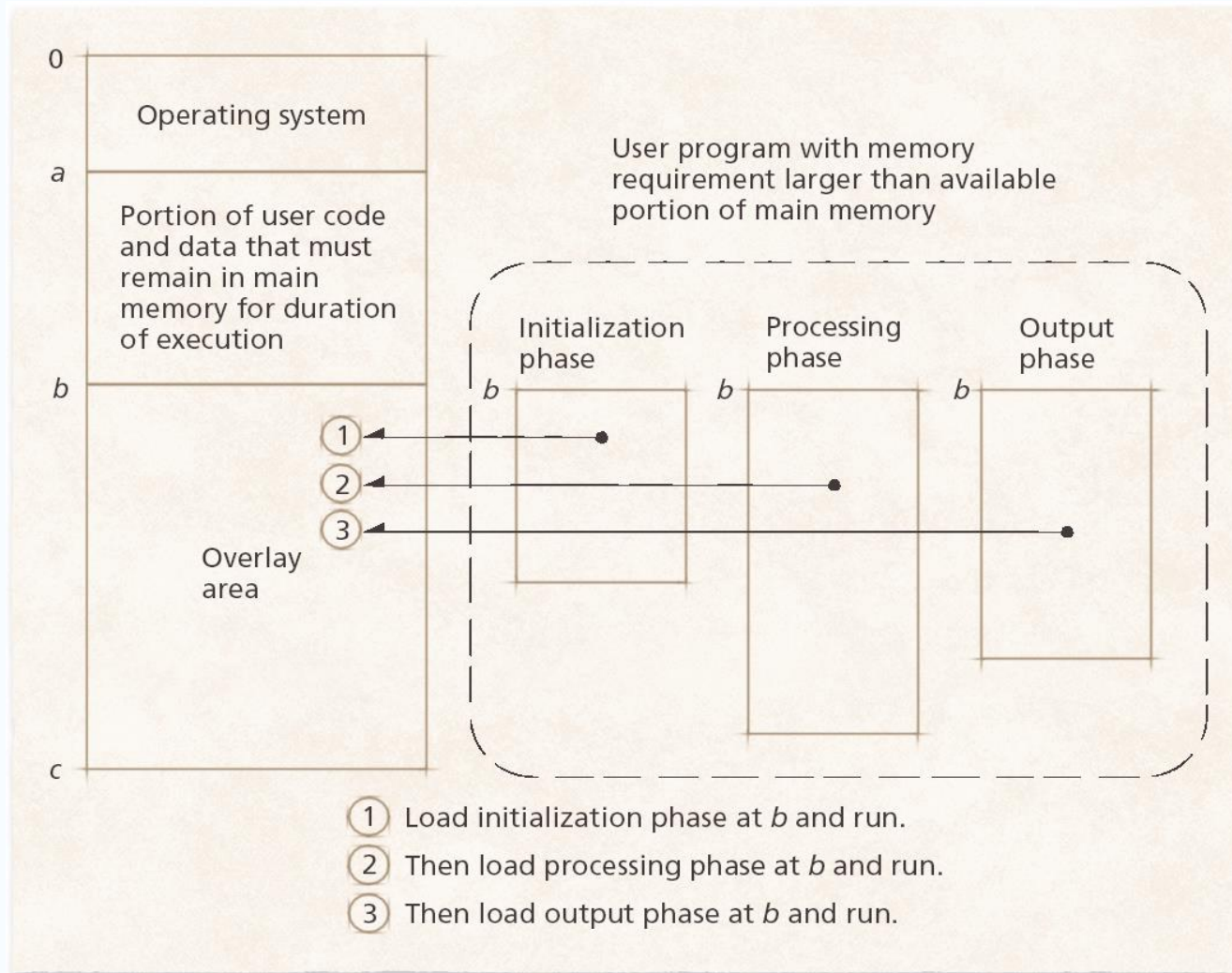


0xFFFF

0



Επικαλύψεις (overlays)



Μνήμη και πολυπρογραμματισμός

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



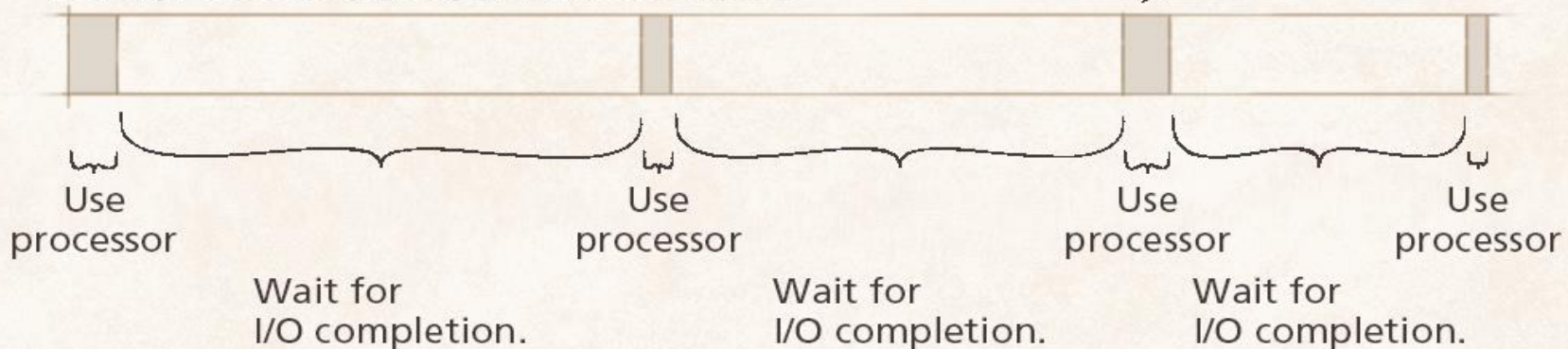
Χρήση της ΚΜΕ σε σύστημα ενός χρήστη

For a process doing intensive calculation:



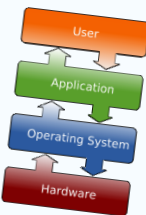
Shaded area indicates
"Processor in use."

For a process doing regular input/output:

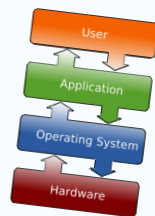
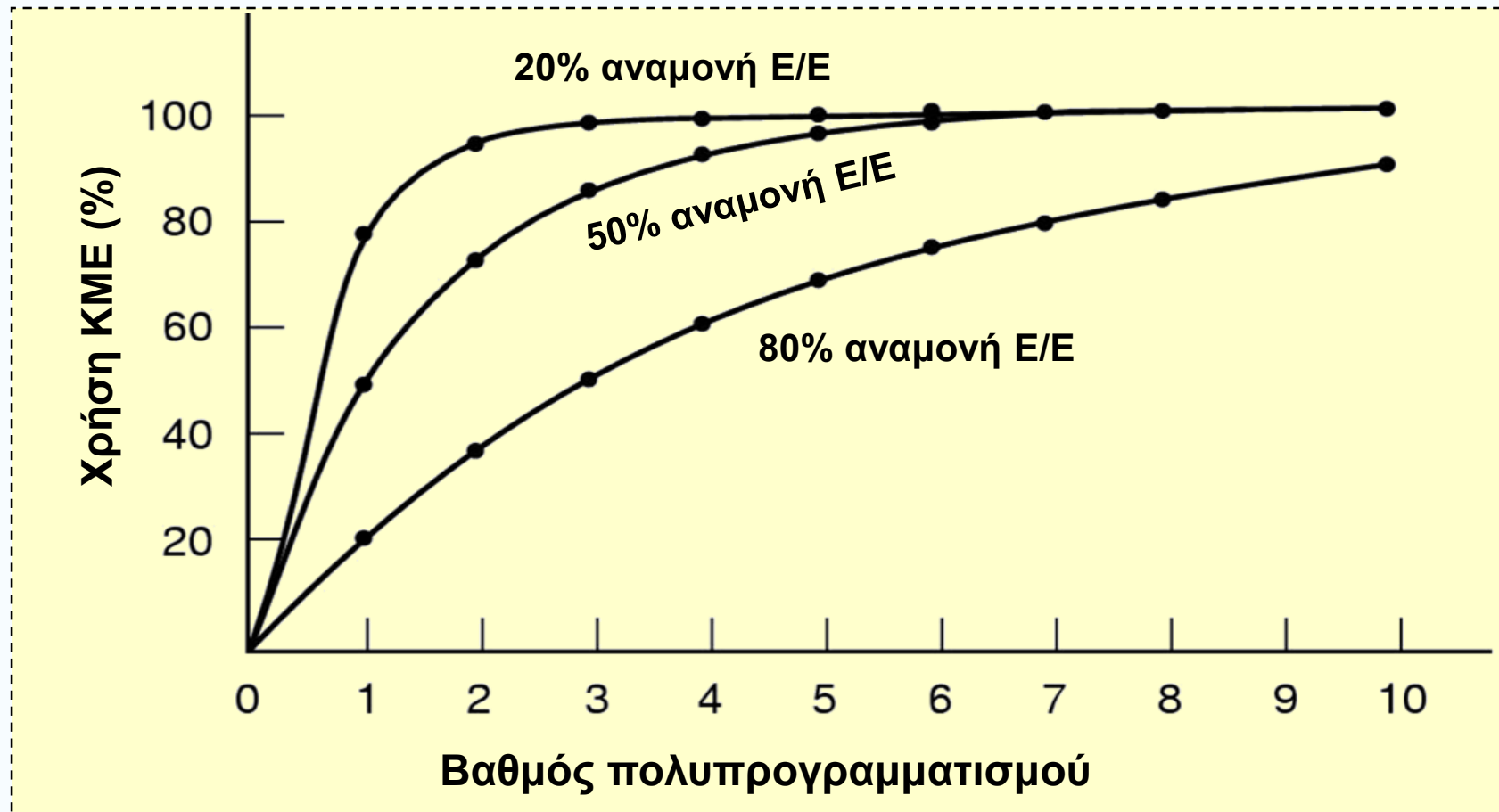


Πόσα προγράμματα είναι αρκετά;

- 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 ΚΜΕ αλλά απαιτείται καλύτερη διαχείριση και προστασία της μνήμης
 - Λιγότερες διεργασίες χρησιμοποιούν λιγότερη μνήμη (φθηνότερα!)
 - Περισσότερη αναμονή για Ε/Ε σημαίνει μικρότερη χρήση επεξεργαστή



Χρήση της ΚΜΕ ως συνάρτηση του πλήθους διεργασιών στη μνήμη



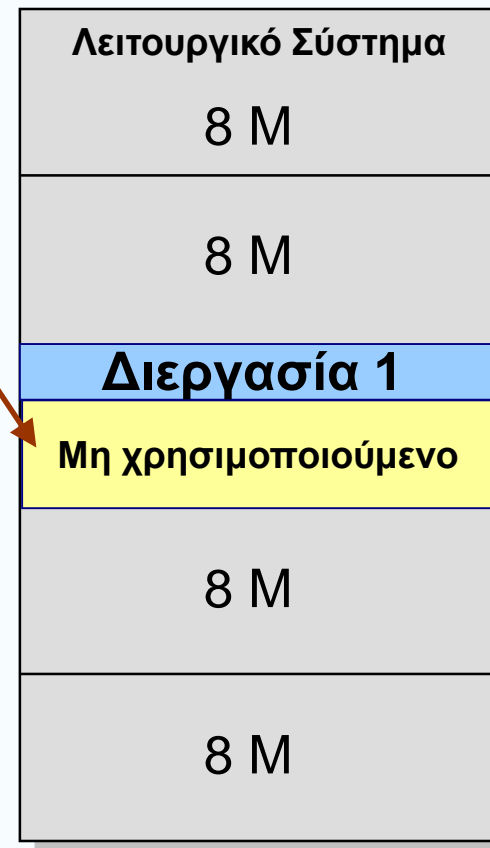
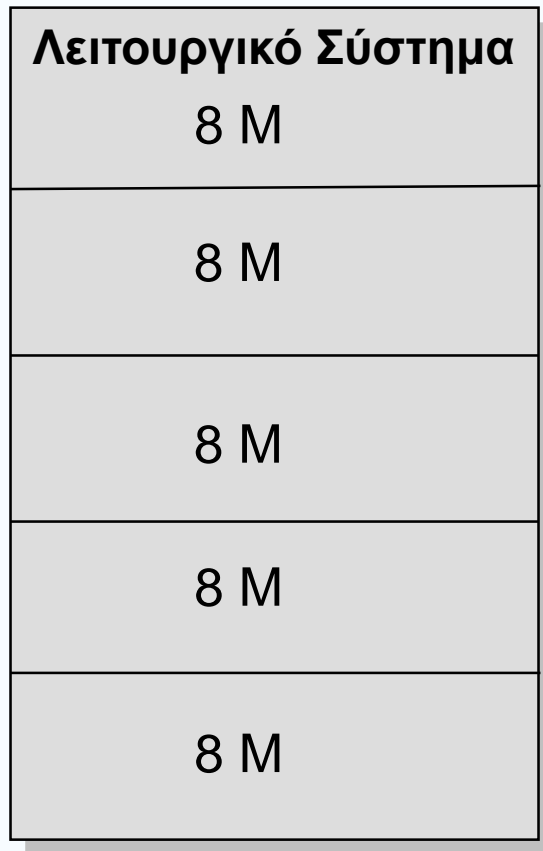
Τμηματοποίηση σταθερού μεγέθους (Fixed Partitioning) – ίσα τμήματα

- 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 ΛΣ 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 χρησιμοποίηση της κύριας μνήμης είναι εξαιρετικά αναποτελεσματική. Κάθε πρόγραμμα, όσο μικρό και να είναι, καταλαμβάνει ένα ολόκληρο τμήμα.
- Ο ανεκμετάλλευστος χώρος εσωτερικά σε ένα τμήμα αναφέρεται ως **εσωτερικός κατακερματισμός (internal fragmentation)**.



Εσωτερικός Κατακερματισμός

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



Εσωτερικός Κατακερματισμός (συν.)

- **Πλεονεκτήματα**

- Μικρή επιβάρυνση στο ΛΣ

- **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

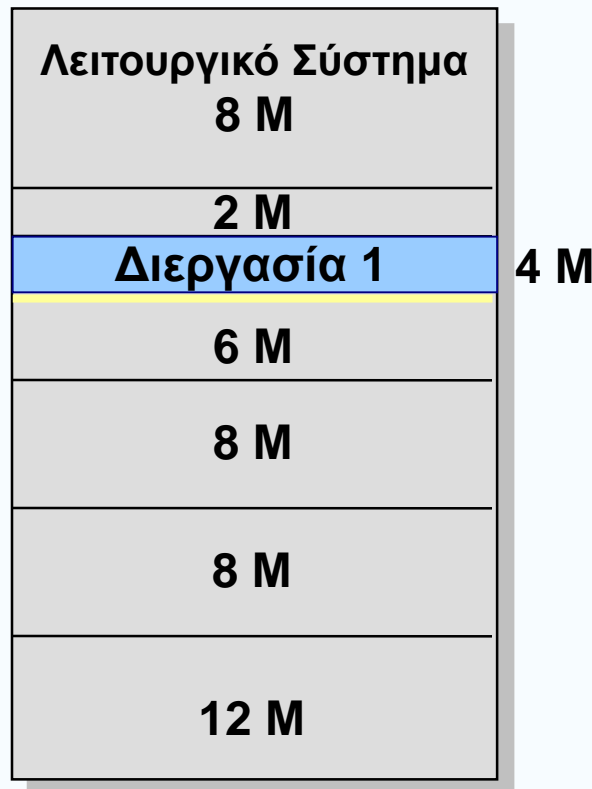


Τμηματοποίηση σταθερού μεγέθους – άνισα τμήματα

Μειώνει τα προβλήματα της τμηματοποίησης ίσων τμημάτων

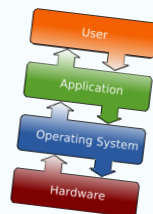
Πλεονεκτήματα

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 δεν φαίνονται στο χρήστη και δεν χρησιμοποιούνται λόγω μεγέθους).



Κατακερματισμός

- **ΕΣΩΤΕΡΙΚΟΣ**

- 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**

- 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

- Διαφορετικού μεγέθους τμήματα

1. **Ουρά για κάθε τμήμα**

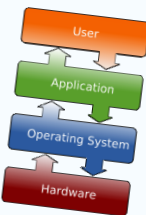
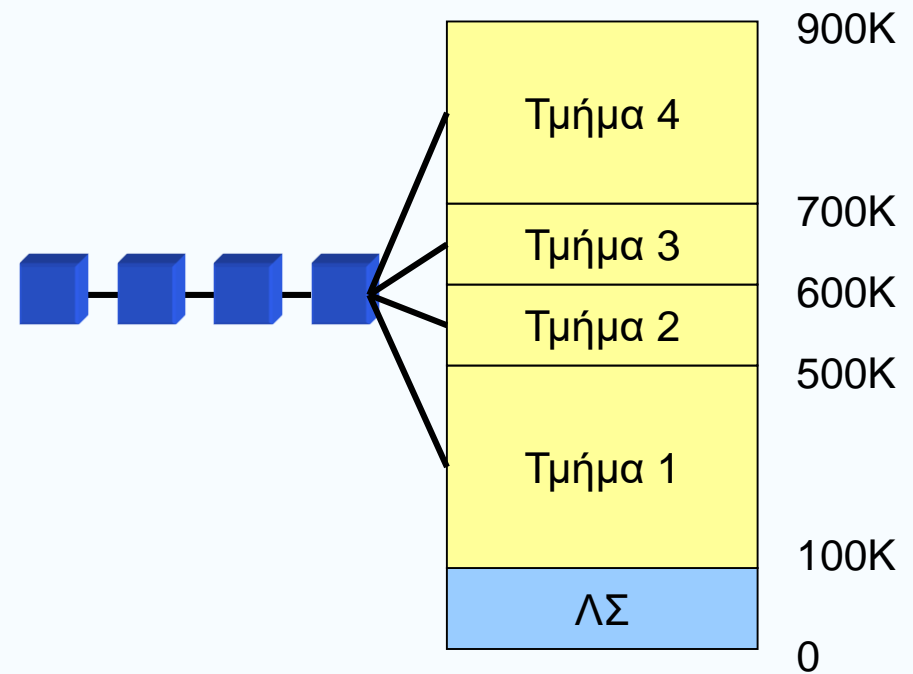
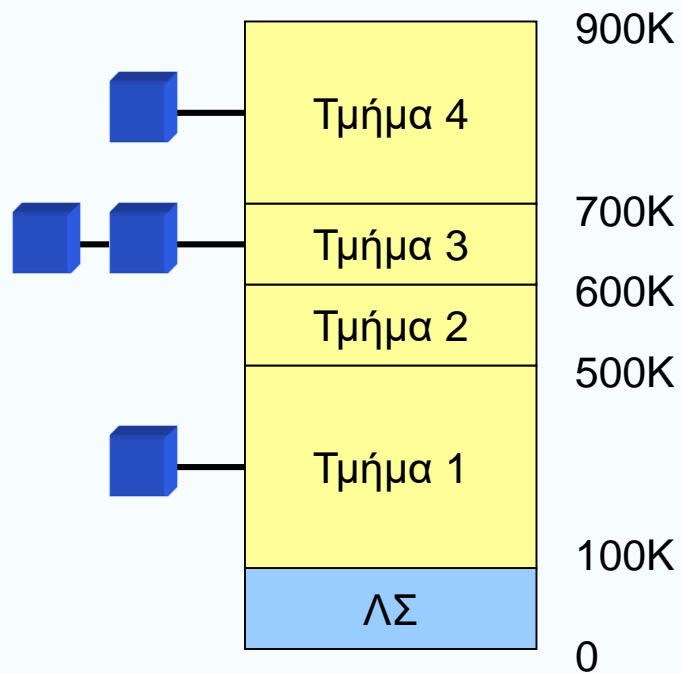
- ◆ Κάθε διεργασία μπορεί να αντιστοιχηθεί στο μικρότερο τμήμα στο οποίο χωράει.
- ◆ Οι διεργασίες αντιστοιχίζονται με τέτοιο τρόπο, ώστε να ελαχιστοποιείται η σπατάλη μνήμης μέσα σε ένα τμήμα (εσωτερικός κατακερματισμός)

2. **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 ΚΜΕ

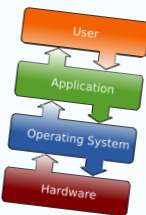


Μια και πολλές ουρές...

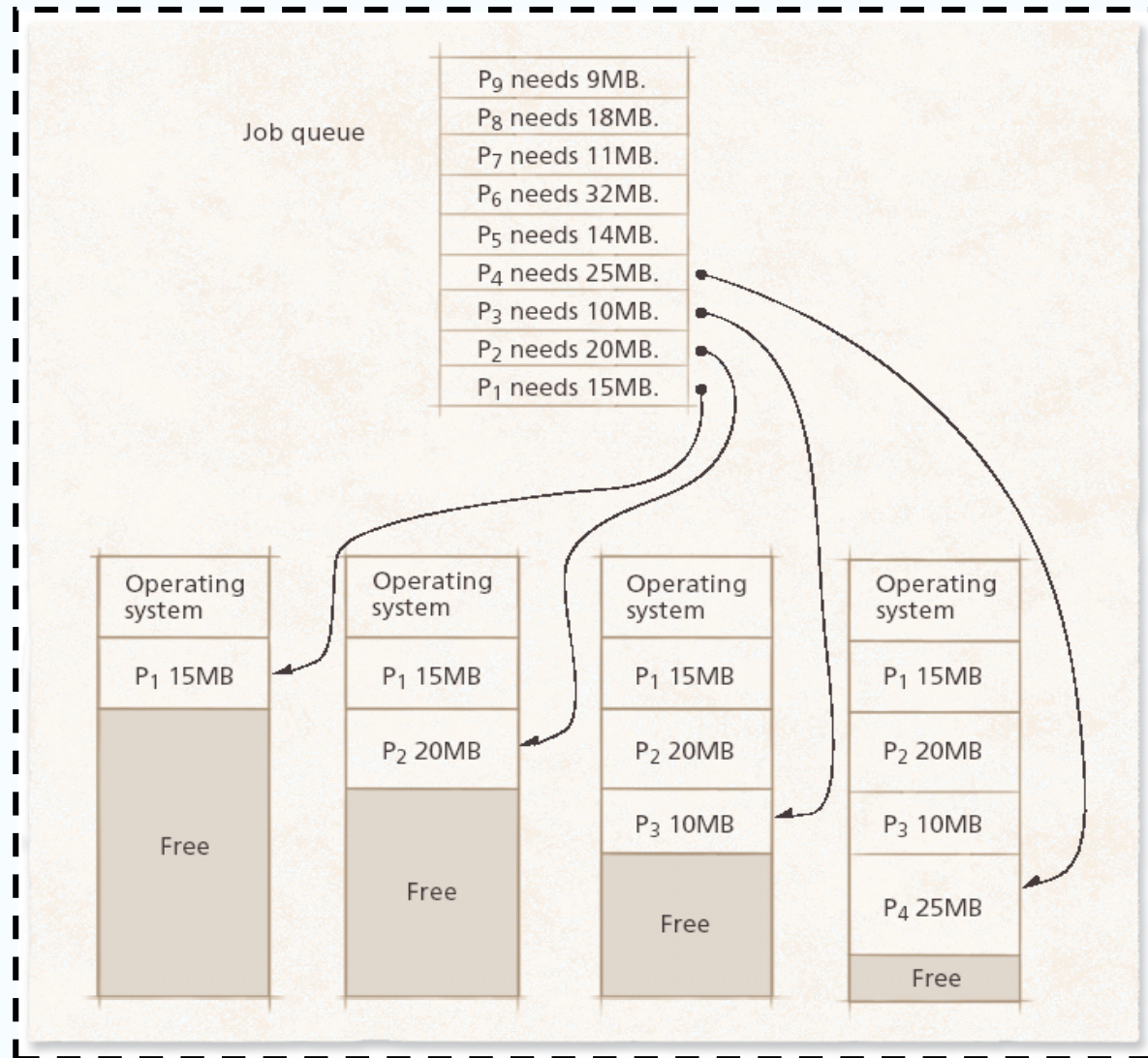


Δυναμική τμηματοποίηση

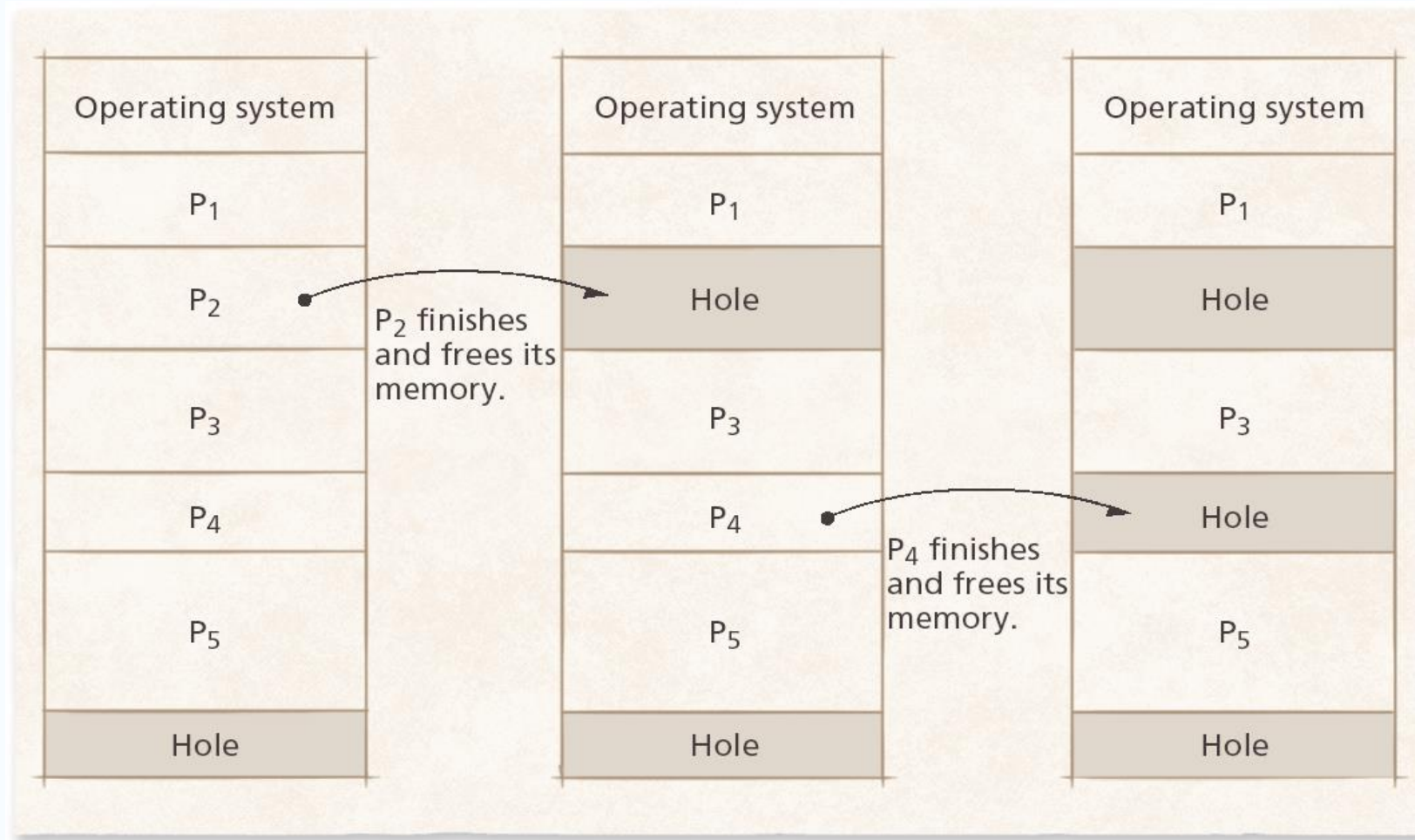
- Τμήματα μεταβλητού μεγέθους και πλήθους
- Μια διεργασία αντιστοιχείται ακριβώς στην ποσότητα μνήμης που απαιτείται
- Τελικά υπάρχουν κενά στη μνήμη. Αυτός είναι ο **εξωτερικός κατακερματισμός (external fragmentation)**
- Πρέπει να χρησιμοποιηθεί συμπίεση (compaction) που θα μετατοπίσει τις διεργασίες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



Δυναμική τμηματοποίηση (συν.)



Εξωτερικός κατακερματισμός



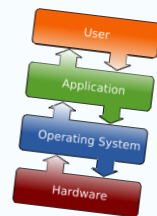
Αλγόριθμοι τοποθέτησης δυναμικής τμηματοποίησης

- Το ΛΣ πρέπει να αποφασίσει ποιο ελεύθερο τμήμα της μνήμης θα εκχωρήσει σε μια διεργασία
- **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



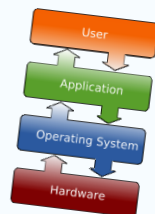
Αλγόριθμοι τοποθέτησης δυναμικής τμηματοποίησης (συν.)

- **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



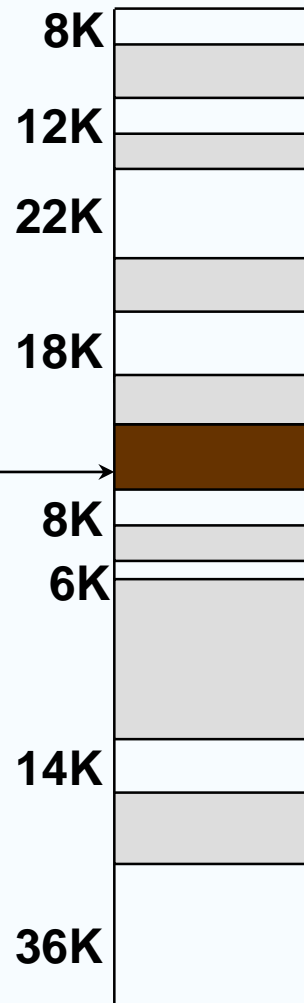
Αλγόριθμοι τοποθέτησης δυναμικής τμηματοποίησης (συν.)

- **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.



Εκχώρηση block 16K

Τελευταία
Δεσμευμένο
block (14K)



Πριν

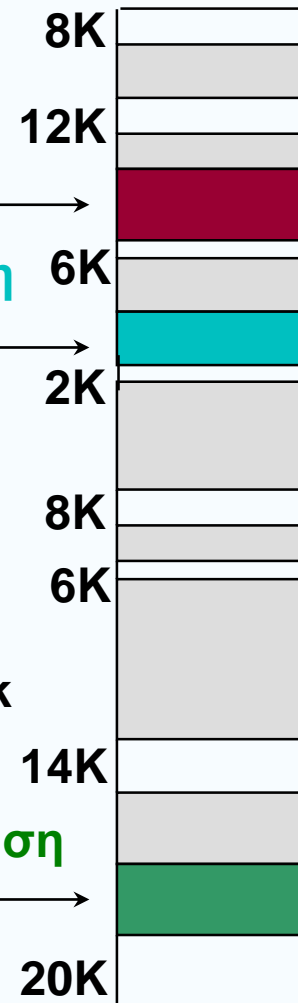
Πρώτη τοποθέτηση

Βέλτιστη τοποθέτηση

Δεσμευμένο block

Ελεύθερο block

Επόμενη τοποθέτηση



Μετά



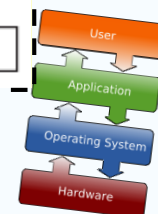
Σύστημα ζευγών ή φίλων

- 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 το block
 - 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

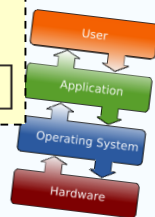
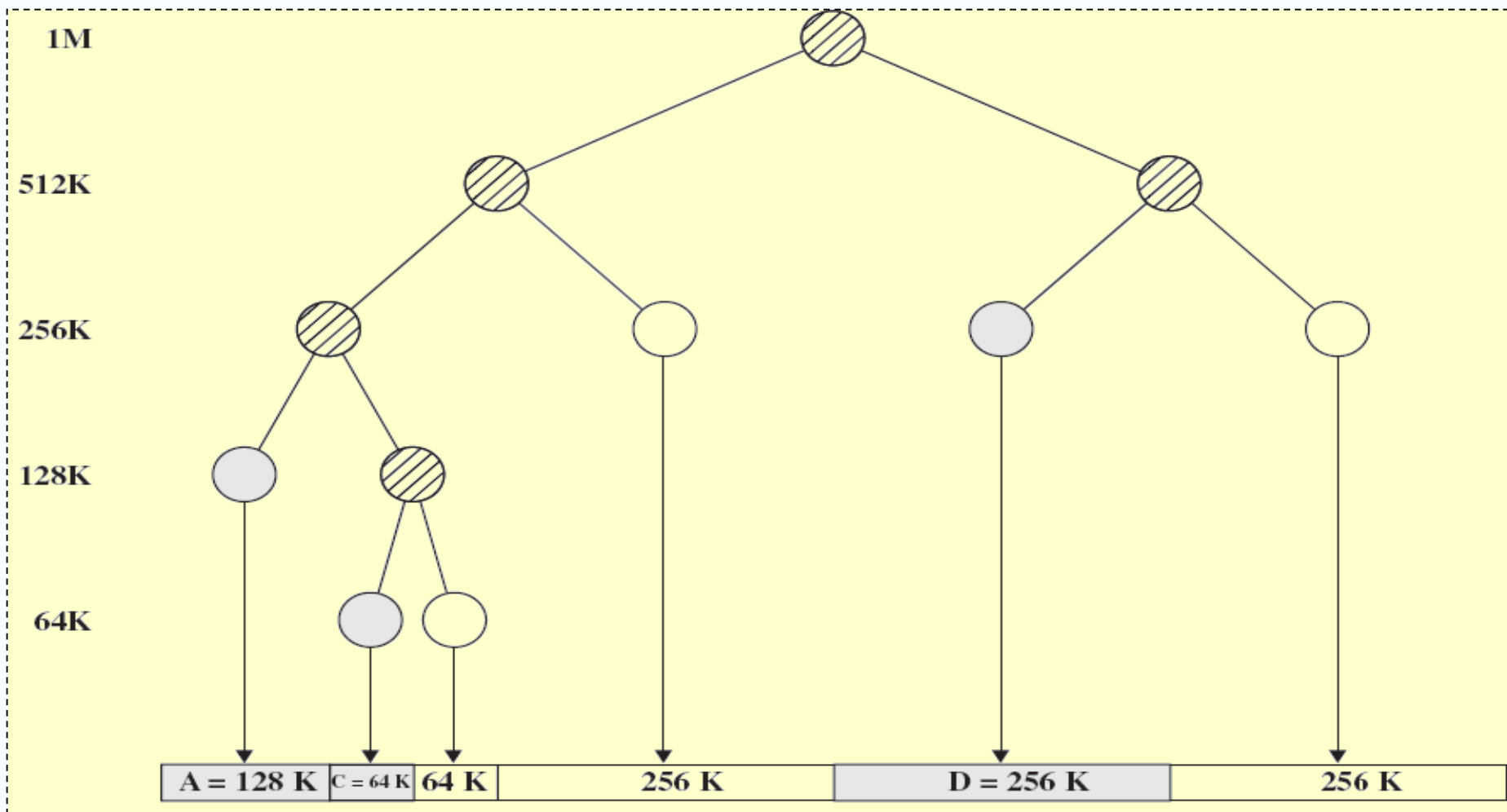


Παράδειγμα

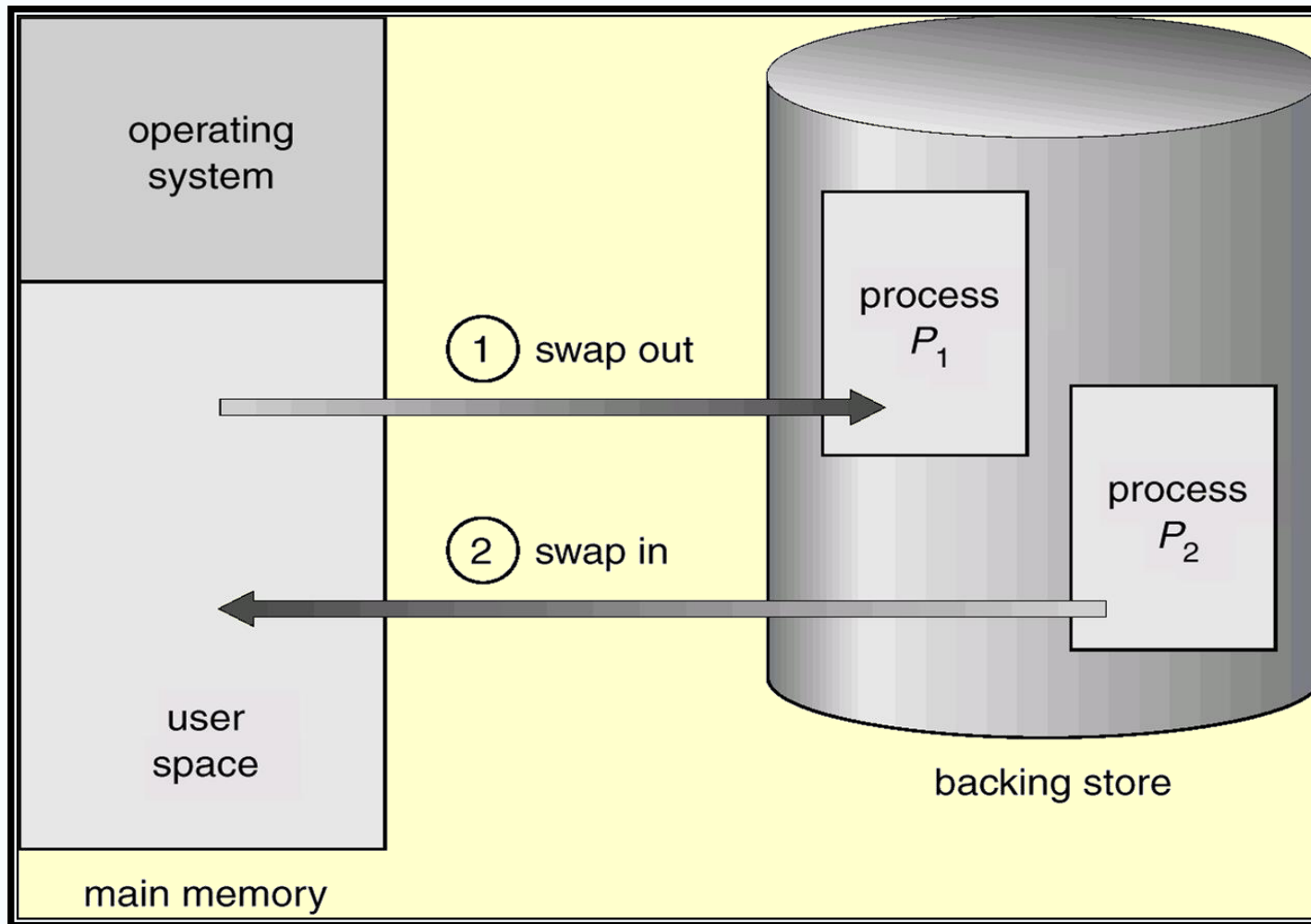
1 Mbyte block	1 M				
Request 100 K	A = 128 K	128 K	256 K	512 K	
Request 240 K	A = 128 K	128 K	B = 256 K	512 K	
Request 64 K	A = 128 K	C = 64 K	64 K	B = 256 K	512 K
Request 256 K	A = 128 K	C = 64 K	64 K	B = 256 K	D = 256 K
Release B	A = 128 K	C = 64 K	64 K	256 K	D = 256 K
Release A	128 K	C = 64 K	64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



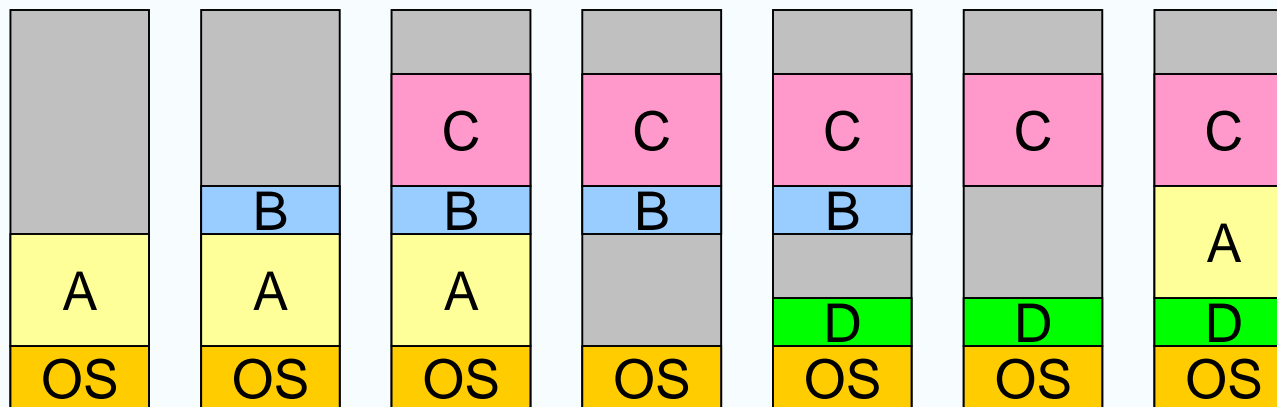
Δενδρική αναπαράσταση του συστήματος ζευγών



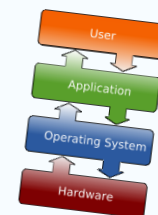
Εναλλαγή (swapping)



Εναλλαγή (Swapping)

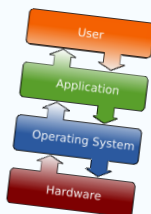
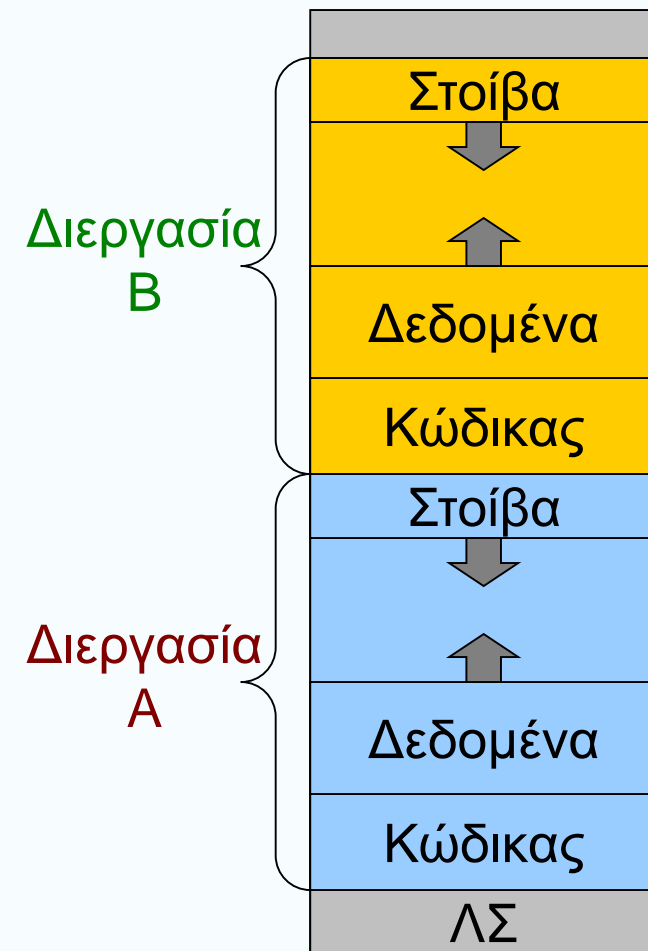


- Η εκχώρηση μνήμης αλλάζει καθώς
 - Έρχονται νέες διεργασίες στη μνήμη
 - Διεργασίες εγκαταλείπουν τη μνήμη
 - ◆ Εναλλάσσονται στο δίσκο
 - ◆ Ολοκληρώνουν την εκτέλεσή τους
- Οι γκρι περιοχές είναι μνήμη που δεν χρησιμοποιείται



Εναλλαγή : αφήνοντας χώρο για ανάπτυξη

- Ανάγκη για τη δυνατότητα ανάπτυξης των προγραμμάτων
 - Εκχώρηση περισσότερης μνήμης για δεδομένα
 - Μεγαλύτερη στοίβα
- 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

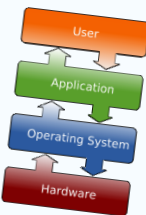
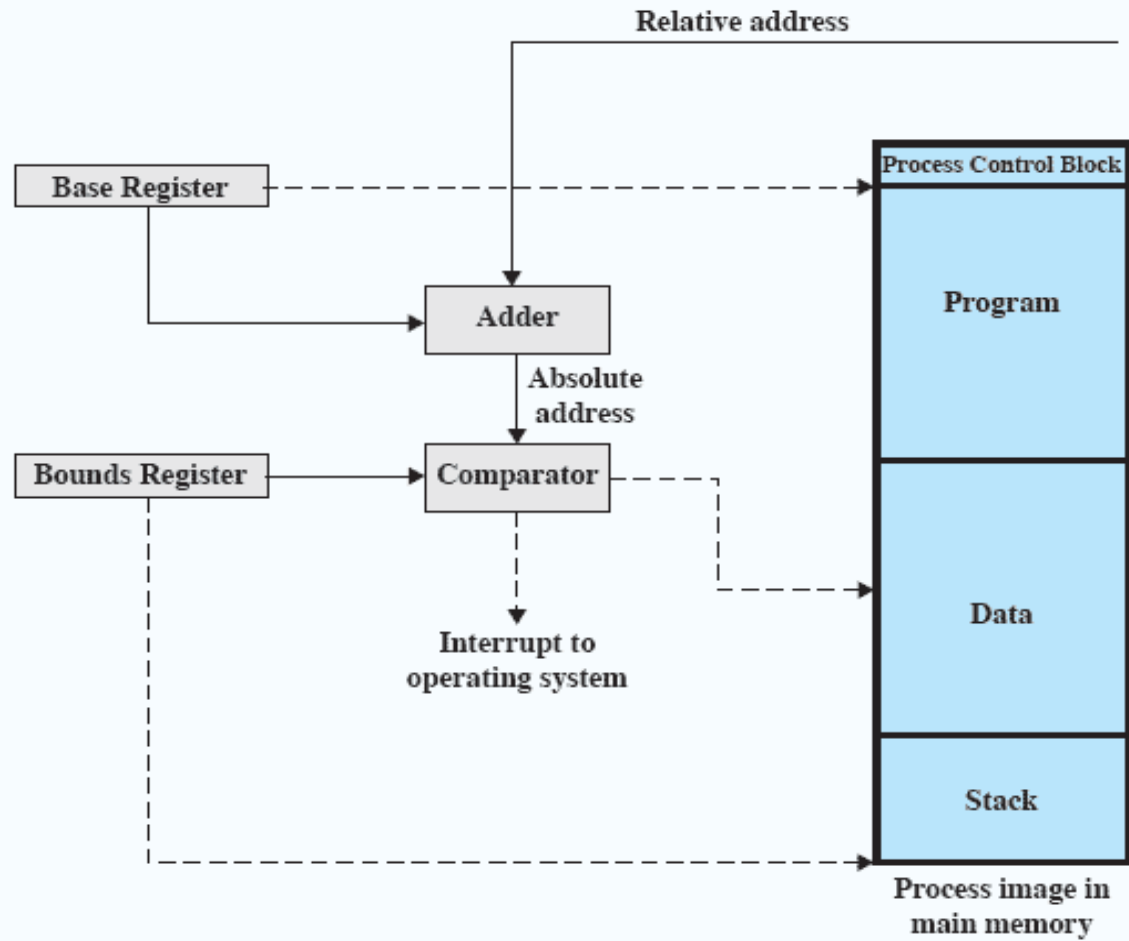
- Προβλήματα με την εναλλαγή
 - Η διεργασία πρέπει να χωρά στη φυσική μνήμη (αδύνατη η εκτέλεση μεγαλύτερων διεργασιών)
 - Η μνήμη κατακερματίζεται (fragmented)
 - ◆ Εξωτερικός κατακερματισμός: πλήθος μικρών ελεύθερων περιοχών
 - ◆ Απαιτείται συμπίεση για την επανασυναρμολόγηση μεγαλύτερων ελεύθερων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



- Λογική
 - Αναφορά σε μια θέση μνήμης ανεξάρτητη με την τρέχουσα ανάθεση των δεδομένων στην κύρια μνήμη
 - Πρέπει να γίνει μετασχηματισμός σε φυσική διεύθυνση
- Σχετική
 - Η διεύθυνση εκφρασμένη ως θέση σχετική με κάποιο δεδομένο (γνωστό) σημείο
- Φυσική
 - Η απόλυτη διεύθυνση ή πραγματική θέση στην κύρια μνήμη

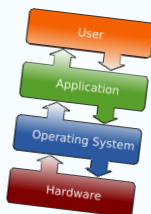


Μετατόπιση



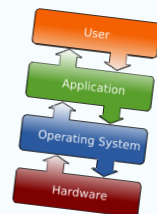
Καταχωρητές σε χρήση κατά την εκτέλεση διεργασίας

- Καταχωρητής βάσης (Base register)
 - Η αρχική διεύθυνση της διεργασίας
- Καταχωρητής ορίων (Bounds register)
 - Η τελική θέση της διεργασίας
- Οι τιμές αυτές ορίζονται όταν η διεργασία φορτώνεται ή έρχεται από εναλλαγή
- Η τιμή του καταχωρητή βάσης προστίθεται στη σχετική διεύθυνση για να παραχθεί η απόλυτη διεύθυνση
- Η διεύθυνση που υπολογίζεται συγκρίνεται με την τιμή στον καταχωρητή ορίων
- Αν είναι εκτός ορίων, παράγεται μια διακοπή στο ΛΣ



Σελιδοποίηση

- Διαχωρισμός της μνήμης σε μικρά, ισομεγέθη τμήματα και διαίρεση κάθε διεργασίας σε ίδιου μεγέθους τμήματα
- Τα τμήματα της διεργασίας ονομάζονται **σελίδες** (pages) και τα τμήματα της μνήμης **πλαίσια** (frames)
- Το ΛΣ διατηρεί έναν πίνακα σελίδων για κάθε διεργασία
 - Περιέχει τη θέση του πλαισίου κάθε σελίδας της διεργασίας
 - Η διεύθυνση μνήμης αποτελείται από έναν αριθμό σελίδας και έναν ρυθμιστή μέσα σε αυτή



Διεργασίες και Πλαίσια

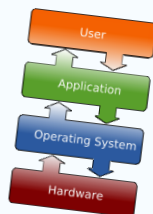
Αριθμός πλαισίου	Κύρια Μνήμη	Κύρια Μνήμη	Κύρια Μνήμη
0		A.0	A.0
1		A.1	A.1
2		A.2	A.2
3		A.3	A.3
4			B.0
5			B.1
6			B.2
7			
8			
9			
10			
11			
12			
13			
14			



Διεργασίες και Πλαίσια (συν.)



Αριθμός πλαisiού	Κύρια Μνήμη	Κύρια Μνήμη	Κύρια Μνήμη
0	A.0	A.0	A.0
1	A.1	A.1	A.1
2	A.2	A.2	A.2
3	A.3	A.3	A.3
4	B.0		D.0
5	B.1		D.1
6	B.2		D.2
7	C.0	C.0	C.0
8	C.1	C.1	C.1
9	C.2	C.2	C.2
10	C.3	C.3	C.3
11			D.3
12			D.4
13			
14			



Πίνακας Σελίδων

0	0
1	1
2	2
3	3

Πίνακας Σελίδων
Διεργασίας A

0	—
1	—
2	—

Πίνακας Σελίδων
Διεργασίας B

0	7
1	8
2	9
3	10

Πίνακας Σελίδων
Διεργασίας C

0	4
1	5
2	6
3	11
4	12

Πίνακας Σελίδων
Διεργασίας D

13
14

Πίνακας ελεύθερων
πλαισίων

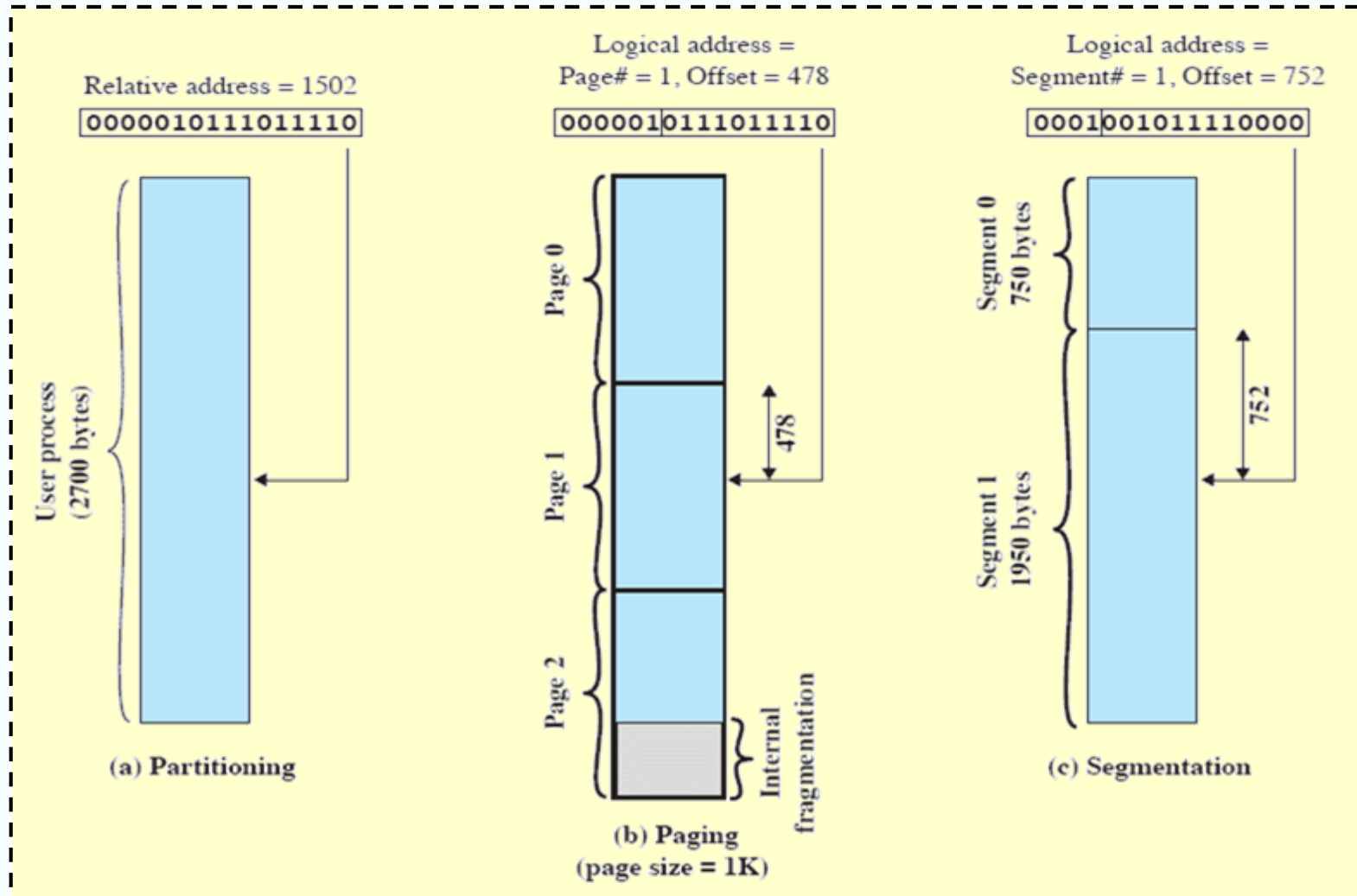


Κατάτμηση

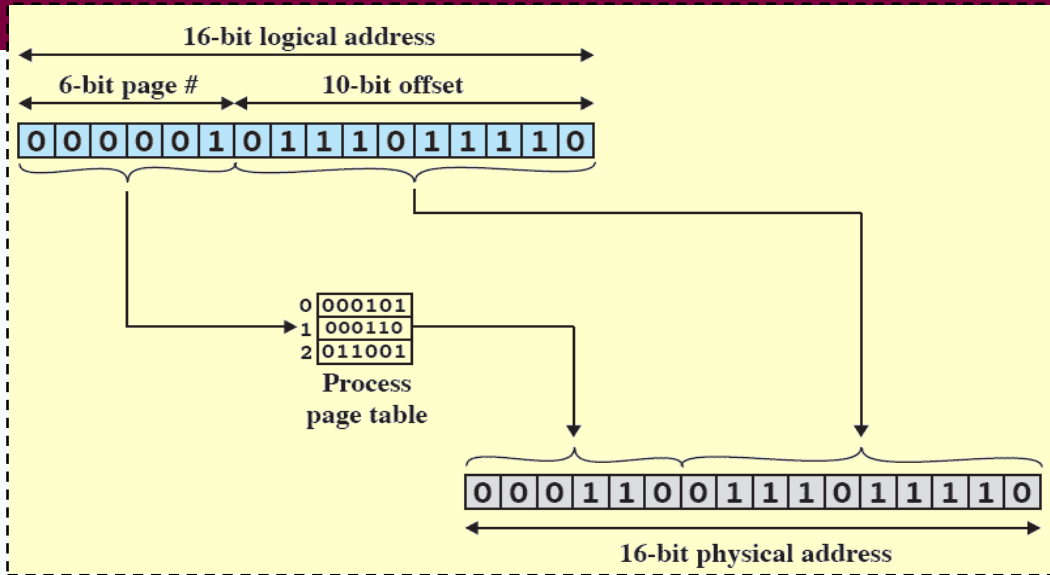
- Τα τμήματα των προγραμμάτων δεν είναι ίσου μεγέθους
- Υπάρχει ένα μέγιστο μήκος τμήματος
- Η διευθυνσιοδότηση αποτελείται από δυο τμήματα – τον αριθμό τμήματος και έναν αντισταθμιστή
- Καθώς τα τμήματα είναι άνισα, η κατάτμηση είναι παρόμοια με δυναμικό διαμερισμό



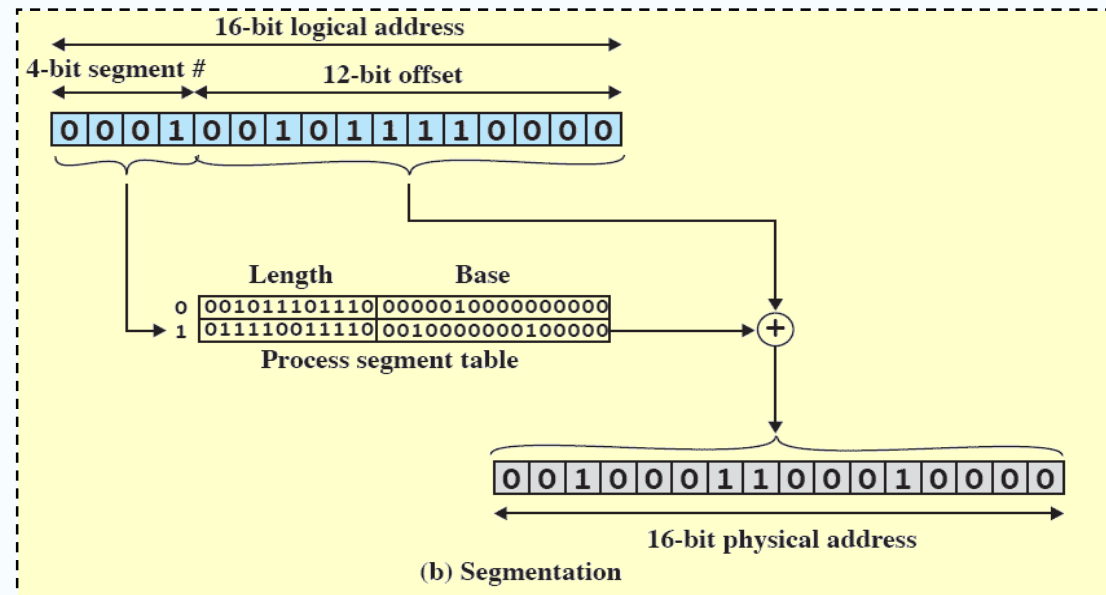
Λογική Διεύθυνση



Σελιδοποίηση



Κατάτμηση



Άσκηση 7.1

- Ένα τμήμα μήκους 1Mbyte ανατίθεται χρησιμοποιώντας το σύστημα ζεύγους φίλων.
- **Δείξτε σχηματικά** τα αποτελέσματα της επόμενης ακολουθίας:
 - Request 70; Request 35; Request 80; Return A; Request 60; Return B; Return D; Return C;
- **Σχεδιάστε** το δυαδικό δένδρο αναπαράστασης που ακολουθεί την Return B



Άσκηση 7.2

- 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, and 426KB.
- **Δείξτε πώς** ο αλγόριθμος πρώτης τοποθέτησης θα διευθετήσει τις απαιτήσεις στην ελεύθερη μνήμη. Υπολογίστε το ελεύθερο τμήμα μνήμης μετά από κάθε άφιξη.
- **Δείξτε πώς** ο αλγόριθμος βέλτιστης τοποθέτησης θα διευθετήσει τις ίδιες απαιτήσεις στην ελεύθερη μνήμη. Υπολογίστε το ελεύθερο τμήμα μνήμης μετά από κάθε άφιξη.
- **Δείξτε πώς** 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 7.3

- 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 αχρησιμοποίητα 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 7.4

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:

1. ποιο τμήμα θα επιλεγεί για απαίτηση 1603 bytes με τη μέθοδο best-fit;
2. ποιο τμήμα θα επιλεγεί για απαίτηση 949 bytes με τη μέθοδο best-fit;
3. ποιο τμήμα θα επιλεγεί για απαίτηση 963 bytes με τη μέθοδο first-fit;
4. ποιο τμήμα θα επιλεγεί για απαίτηση 349 bytes με τη μέθοδο first-fit;
5. 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;



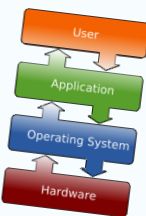
Άσκηση 7.5

- 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;



Βασικά σημεία

- Η διαχείριση μνήμης αφορά στην αντιμετώπιση της κύριας μνήμης ως έναν πόρο που πρέπει να ανατεθεί και να διαμοιραστεί σε έναν αριθμό από διεργασίες.
- Απώτερος στόχος είναι η όσο το δυνατόν αποδοτικότερη χρήση του ΚΜΕ και των συσκευών Ε/Ε -> πρέπει να υπάρχουν όσο το δυνατόν περισσότερες διεργασίες στη μνήμη.
- Βασικά εργαλεία για τη διαχείριση της μνήμης είναι:
 - Η τμηματοποίηση (σταθερού ή μεταβλητού μήκους)
 - Η σελιδοποίηση
 - Η κατάτμηση



Αναφορές

- “Λειτουργικά Συστήματα – Αρχές Σχεδίασης”, 4η έκδοση, W. Stallings, Εκδόσεις Τζιόλα, 2008.
- “Operating System Concepts”, 7η έκδοση, από Abraham Silberschatz, Peter Galvin και Greg Gagne, Addison-Wesley, 2004.
- “Operating Systems: Design and Implementation”, 3η έκδοση, από Andrew Tanenbaum και Albert Woodhull, Prentice Hall, 2006.

