ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Λειτουργικά Συστήματα

Ενότητα # 3: Διαχείριση Μνήμης

Διδάσκων: Γεώργιος Ξυλωμένος

Τμήμα: Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Οικονομικό Πανεπιστήμιο Αθηνών**» 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

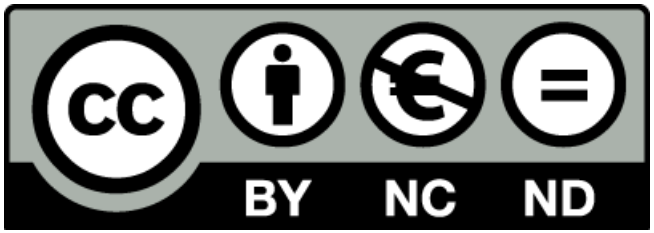
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Οι εικόνες προέρχονται από το βιβλίο «Σύγχρονα Λειτουργικά Συστήματα», A.S. Tanenbaum, 3η έκδοση, 2009, Εκδόσεις Κλειδάριθμος.



Σκοποί ενότητας

- Κατανόηση της ανάγκης αφαίρεσης μνήμης
- Εξοικείωση με τις βασικές τεχνικές οργάνωσης της εικονικής μνήμης: σελιδοποίηση και τμηματοποίηση
- Κατανόηση των βασικών αλγορίθμων αντικατάστασης σελίδων
- Εξοικείωση με τα βασικά ζητήματα σχεδιασμού και υλοποίησης εικονικής μνήμης

Περιεχόμενα ενότητας

- Χωρίς αφαίρεση μνήμης
- Χώροι διευθύνσεων
- Εικονική μνήμη
- Αλγόριθμοι αντικατάστασης σελίδων
- Θέματα σχεδιασμού
- Ζητήματα υλοποίησης
- Τμηματοποίηση

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Εισαγωγή

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 3:** Διαχείριση Μνήμης

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Εισαγωγή (1 από 2)

- Η μνήμη είναι μόνιμα κρίσιμος πόρος
  - Η διαθέσιμη μνήμη αυξάνεται συνεχώς
  - Δυστυχώς μεγαλώνουν και τα προγράμματα!
  - Ιδανικά: μόνιμη, γρήγορη, άπειρη, φτηνή μνήμη
- Ιεραρχία μνήμης
  - Από την κρυφή μνήμη του επεξεργαστή
  - Μέχρι τους μαγνητικούς δίσκους
  - Διαφορετικοί συμβιβασμοί ταχύτητας / κόστους

Εισαγωγή (2 από 2)

- Διαχειριστής μνήμης
  - Διαχειρίζεται την ιεραρχία μνήμης
  - Ποια τμήματα της μνήμης χρησιμοποιούνται;
  - Πόση μνήμη παραχωρείται σε κάθε διεργασία;
  - Ποια μνήμη έχει κάθε διεργασία;
  - Λογισμικό που αξιοποιεί εξειδικευμένο υλικό
    - Υλικό που παρακολουθεί τις προσπελάσεις
    - Υλικό που επιβάλλει κανόνες προστασίας

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

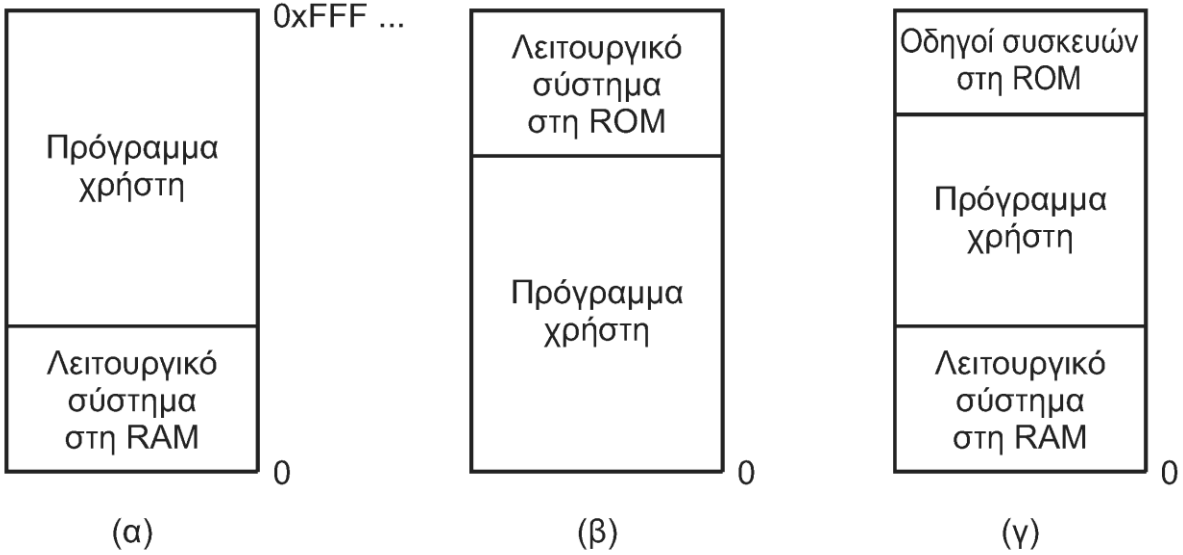
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Χωρίς αφαίρεση μνήμης

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 3:** Διαχείριση Μνήμης

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Χωρίς αφαίρεση μνήμης (1 από 5)



- Όλοι βλέπουν απευθείας τη μνήμη
  - Σύνολο διευθύνσεων από 0 έως n
  - Ένα μόνο πρόγραμμα σε κάθε στιγμή
  - Λειτουργικό σύστημα σε RAM ή ROM

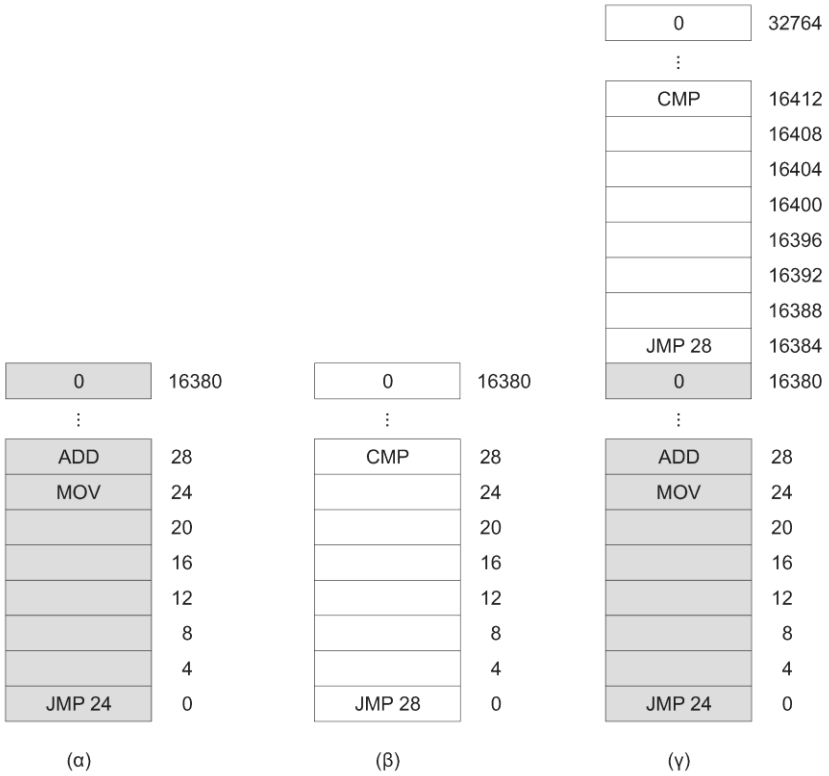
Χωρίς αφαίρεση μνήμης (2 από 5)

- Λειτουργία συστημάτων με μία διεργασία
  - Το λειτουργικό δίνει προτροπή (prompt) στο χρήστη
  - Ο χρήστης πληκτρολογεί το όνομα του προγράμματος
  - Το πρόγραμμα φορτώνεται στη μνήμη και εκτελείται
  - Στο τέλος επιστρέφουμε στο λειτουργικό
  - Χρήση σε πρωτόλεια ή πολύ απλά συστήματα
  - Δυνατότητα χρήσης πολλών νημάτων
  - Πολυπρογραμματισμός με εναλλαγή
    - Φόρτωση και αποθήκευση ολόκληρων διεργασιών

Χωρίς αφαίρεση μνήμης (3 από 5)

- Πολυπρογραμματισμός χωρίς αφαίρεση μνήμης
  - Η συνύπαρξη στη μνήμη απαιτεί πρόσθετο υλικό
  - Παράδειγμα: IBM 360 με κλειδιά προστασίας
    - Κάθε μπλοκ μνήμης των 2 KB έχει ένα κλειδί προστασίας
    - Το εκτελούμενο πρόγραμμα έχει ένα κλειδί προστασίας
  - Η ανάγκη της επανατοποθέτησης (relocation)
    - Έστω ότι έχουμε δύο προγράμματα που χωράνε στη μνήμη
    - Τα προγράμματα δεν ξέρουν πού θα τοποθετηθούν
  - Στατική επανατοποθέτηση στον IBM 360
    - Οι αναφορές σε διευθύνσεις αλλάζουν κατά τη φόρτωση

Χωρίς αφαίρεση μνήμης (4 από 5)



- Παράδειγμα: δύο προγράμματα στη μνήμη
  - Θα πρέπει το ένα να επανατοποθετηθεί

Χωρίς αφαίρεση μνήμης (5 από 5)

- Υλοποίηση της επανατοποθέτησης
  - Η επανατοποθέτηση επηρεάζει ορισμένες εντολές
    - Όσες αναφέρονται σε απόλυτες διευθύνσεις μνήμης
  - Ο φορτωτής πρέπει να διακρίνει αυτές τις εντολές
    - Είτε θα ελέγχει το πρόγραμμα κατά τη φόρτωση
    - Είτε το πρόγραμμα θα περιέχει αυτές τις πληροφορίες
  - Εναλλακτικά, μεταγλώττιση στις σωστές θέσεις
    - Εφικτό μόνο σε ενσωματωμένα συστήματα
  - Γενικότερες λύσεις απαιτούν πρόσθετο υλικό
    - Δυνατότητα δυναμικής επανατοποθέτησης προγραμμάτων

Χώροι διευθύνσεων

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 3:** Διαχείριση Μνήμης

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

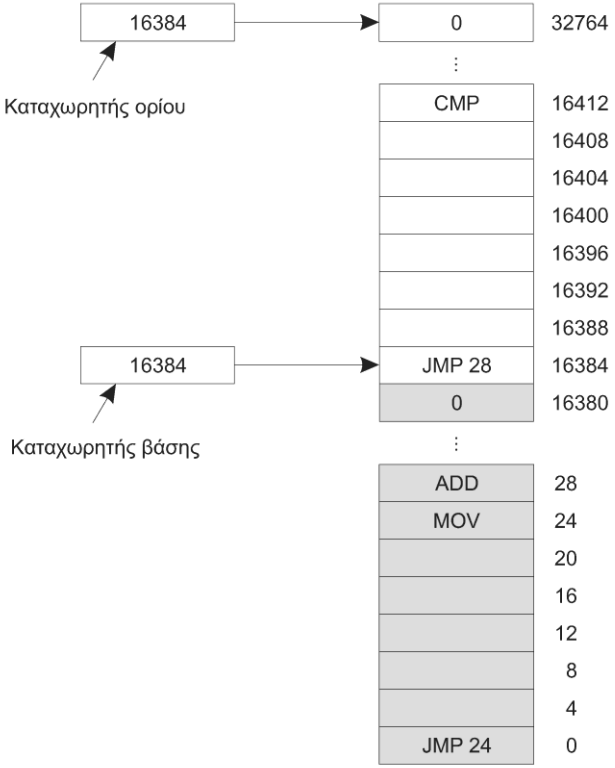
Χώροι διευθύνσεων (1 από 3)

- Γιατί τα προγράμματα να μην βλέπουν τη μνήμη;
  - Μπορούν να διαγράψουν ακόμη και το λειτουργικό
  - Είναι δύσκολο να έχουμε πολλά ταυτόχρονα
- Χώροι διευθύνσεων
  - Λογική αφαίρεση της μνήμης μιας διεργασίας
  - Αποτελείται από το σύνολο των διευθύνσεών της
  - Διάφοροι τρόποι υλοποίησης χώρων διευθύνσεων
- Καταχωρητές βάσης και ορίου
  - Μορφή δυναμικής επανατοποθέτησης
  - Αρχικά σε CDC 6600, παραλλαγή σε 8088/8086

Χώροι διευθύνσεων (2 από 3)

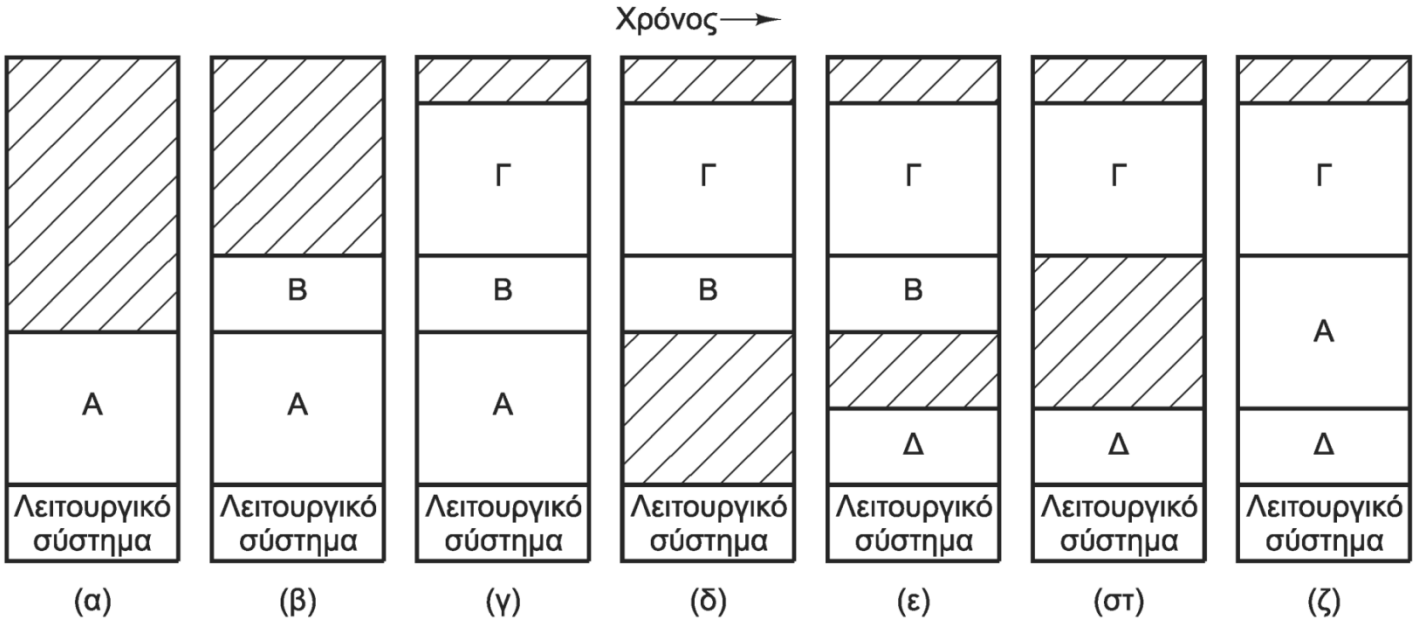
- Καταχωρητές βάσης και ορίου
  - Φόρτωση προγράμματος σε διαδοχικές θέσεις
  - Αρχική διεύθυνση: καταχωρητής βάσης
  - Μέγιστη διεύθυνση: καταχωρητής ορίου
  - Η διεύθυνση δεν πρέπει να περνά το όριο
    - Προστασία μνήμης
  - Η διεύθυνση προστίθεται στη βάση
    - Δυναμική επανατοποθέτηση
- Κόστος πράξεων σε κάθε προσπέλαση
  - Σύγκριση και πρόσθεση

Χώροι διευθύνσεων (3 από 3)



- Παράδειγμα με δύο διεργασίες
  - Οι καταχωρητές αλλάζουν μαζί με τη διεργασία

Εναλλαγή (1 από 4)



- Εναλλαγή διεργασιών
  - Αν δεν χωράνε όλα τα προγράμματα στη μνήμη;
  - Χρήση δίσκου για αποθήκευση αδρανών διεργασιών
  - Εναλλαγή (swapping) των διεργασιών στη μνήμη

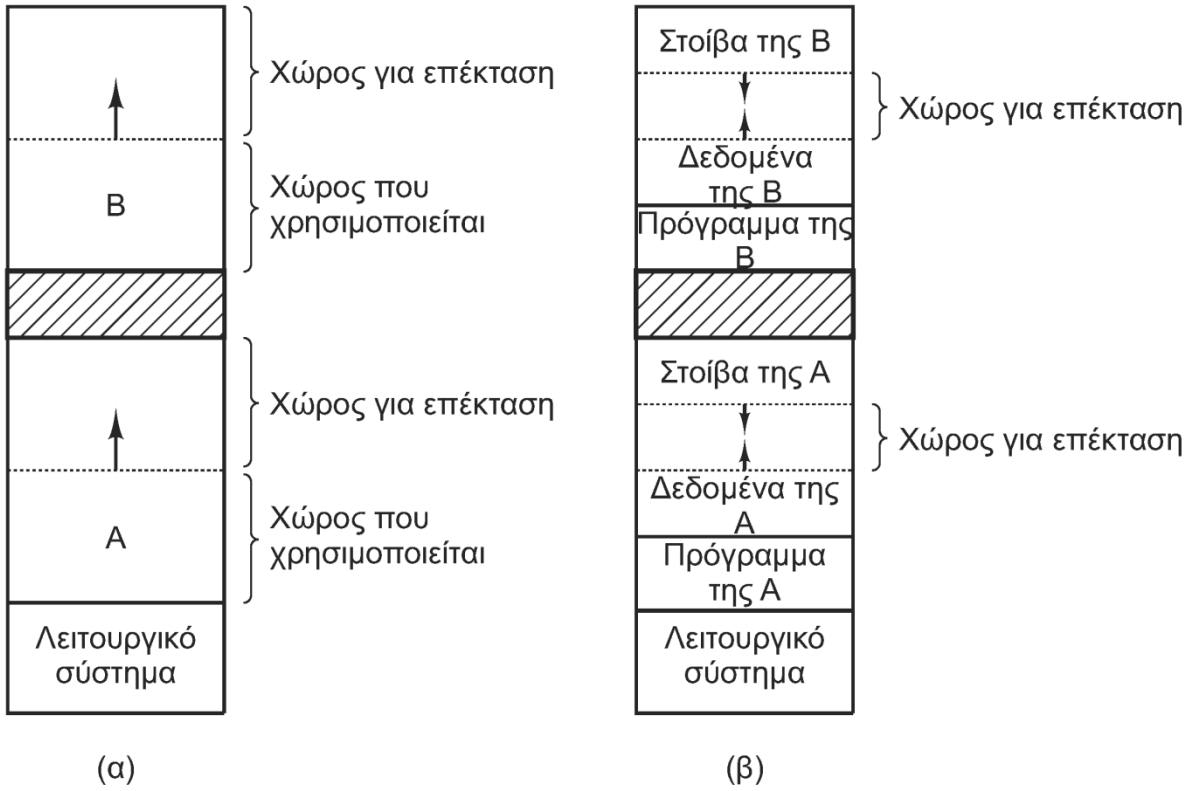
Εναλλαγή (2 από 4)

- Η εναλλαγή δημιουργεί «τρύπες» στη μνήμη
  - Μία διεργασία που φεύγει αφήνει ένα κενό
  - Στη θέση της μπαίνει ίση ή μικρότερη διεργασία
  - Αν είναι μικρότερη, δημιουργείται «τρύπα»
- Περιοδικά χρειάζεται σύμπτυξη μνήμης
  - Μεταφορά διεργασιών για να μην έχουμε κενά
  - Μεγάλο κόστος για την αντιγραφή της μνήμης

Εναλλαγή (3 από 4)

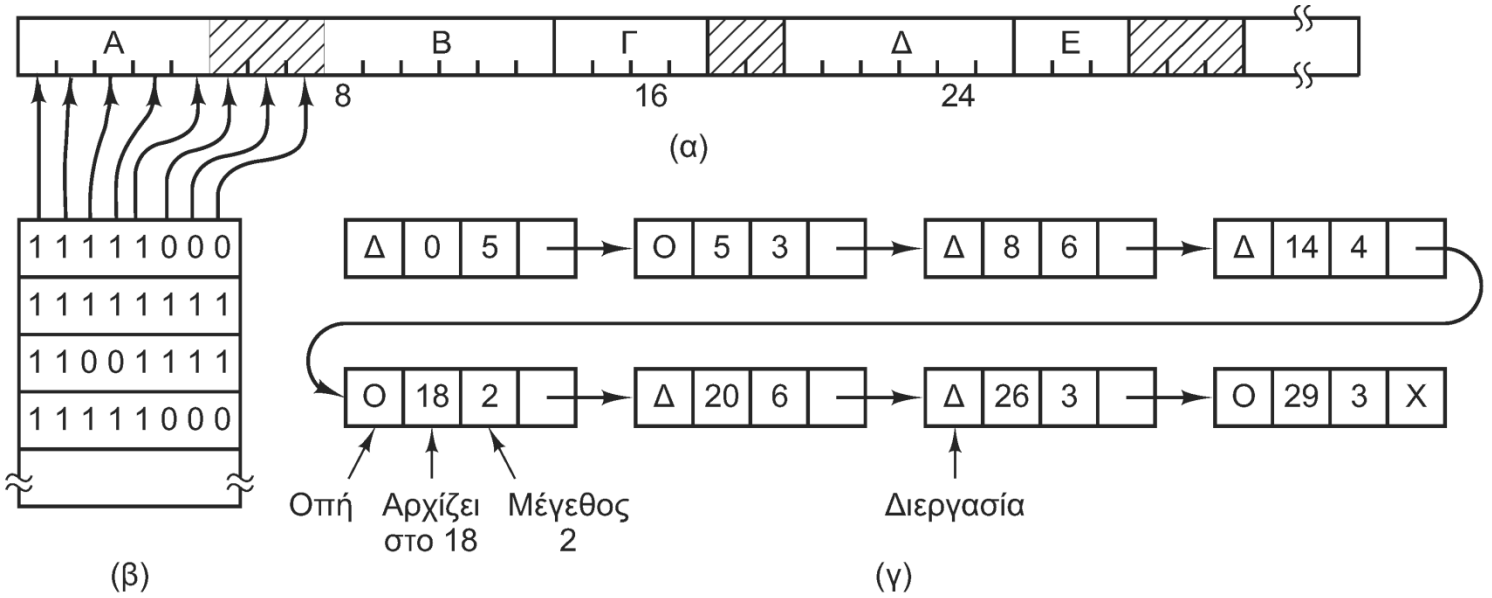
- Διεργασίες με δυναμικό μέγεθος
  - Οι στοίβες και οι σωροί επεκτείνονται δυναμικά
  - Αν περάσουμε την αρχική κατανομή;
  - Θα πρέπει αναγκαστικά να επεκταθούμε
  - Αν η γειτονική περιοχή είναι κενή, ΟΚ
  - Αλλιώς πρέπει να μεταφερθούμε αλλού
  - Γενικά κατανέμουμε χώρο από την αρχή
    - Ο χώρος μπορεί να μείνει αναξιοποίητος

Εναλλαγή (4 από 4)



- Εκχώρηση μνήμης για αύξηση μεγέθους
  - Τα κενά τμήματα δεν μεταφέρονται από/προς το δίσκο

Διαχείριση ελεύθερης μνήμης (1 από 7)



- Το σύστημα παρακολουθεί τη διαθέσιμη μνήμη
- Διαχείριση μνήμης με χάρτες bit
  - Διαίρεση της μνήμης σε ισομεγέθη τμήματα
  - Ένα bit ανά τμήμα δείχνει αν είναι διαθέσιμο

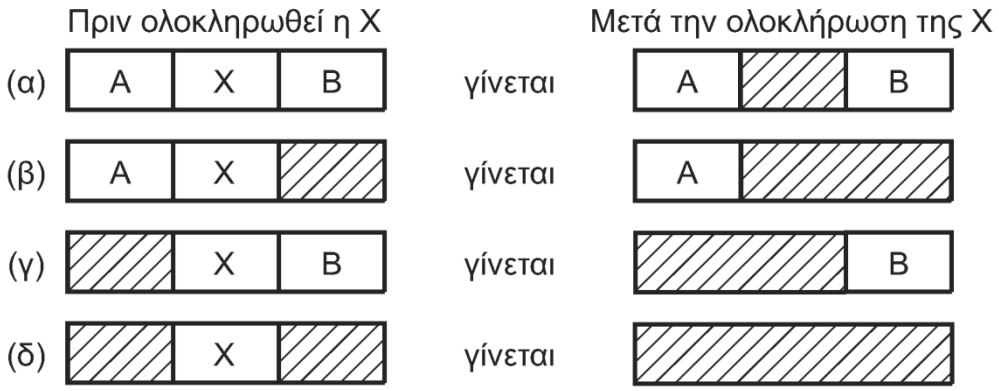
Διαχείριση ελεύθερης μνήμης (2 από 7)

- Διαχείριση μνήμης με χάρτες bit
  - Το μέγεθος της μονάδας κατανομής έχει σημασία
  - Μεγάλη->μικρός χάρτης/μεγάλη σπατάλη
    - Μέρος της τελευταίας μονάδας μπορεί να είναι κενό
  - Μικρή->μεγάλος χάρτης/μικρή σπατάλη
  - Χρονοβόρα αναζήτηση στο χάρτη bit
    - Έστω ότι θέλουμε μια περιοχή k τμημάτων
    - Πρέπει να ψάξουμε σειριακά το χάρτη bit

Διαχείριση ελεύθερης μνήμης (3 από 7)

- Διαχείριση μνήμης με συνδεδεμένες λίστες
  - Λίστα με στοιχεία για κάθε τμήμα μνήμης
    - Τύπος, αρχική διεύθυνση, μήκος
    - Ο τύπος μπορεί να είναι διεργασία ή οπή
- Ταξινόμηση λίστας με διάφορους τρόπους
  - Ταξινόμηση σύμφωνα με τις διευθύνσεις
    - Εύκολη ενημέρωση όταν ελευθερώνεται η μνήμη
    - Γενικά συγχωνεύονται τα γειτονικά στοιχεία

Διαχείριση ελεύθερης μνήμης (4 από 7)



- Διαχείριση μνήμης με συνδεδεμένες λίστες
  - Συνήθως η λίστα είναι διπλά συνδεδεμένη
    - Ενημέρωσή της ξεκινώντας από το τμήμα
    - Η θέση του τμήματος φαίνεται στον πίνακα διεργασιών
  - Διάφορες τεχνικές επιλογής τμήματος προς χρήση

Διαχείριση ελεύθερης μνήμης (5 από 7)

- Επιλογή τμήματος με συνδεδεμένες λίστες
  - Πρώτη προσαρμογή (first fit)
    - Χρησιμοποιείται η πρώτη οπή που ταιριάζει
  - Επόμενη προσαρμογή (next fit)
    - Όπως παραπάνω αλλά συνεχίζει από το τελευταίο σημείο
  - Βέλτιστη προσαρμογή (best fit)
    - Ψάχνουμε όλη τη λίστα για την μικρότερη δυνατή οπή
  - Χειρότερη προσαρμογή (worst fit)
    - Ψάχνουμε όλη τη λίστα για την μεγαλύτερη δυνατή οπή

Διαχείριση ελεύθερης μνήμης (6 από 7)

- Επιλογή τμήματος με συνδεδεμένες λίστες
  - Χρήση χωριστών λιστών για οπές και διεργασίες
    - Πιο γρήγορη αναζήτηση στη λίστα οπών
  - Η λίστα οπών ταξινομείται με το μέγεθος
    - Πιο γρήγορη αναζήτηση της κατάλληλης οπής
    - Βέλτιστη προσαρμογή = πρώτη προσαρμογή
  - Δεν χρειάζεται εξωτερική λίστα οπών
    - Οι δείκτες αποθηκεύονται στις ίδιες τις οπές
  - Πιο δύσκολη ενημέρωση στην αποδέσμευση
    - Πρέπει να δούμε πού θα μπει η διαθέσιμη μνήμη

Διαχείριση ελεύθερης μνήμης (7 από 7)

- Γρήγορη προσαρμογή (quick fit)
  - Διατήρηση λιστών οπών με διάφορα μεγέθη
  - Πολύ γρήγορος εντοπισμός κατάλληλης οπής
  - Πιο δύσκολη ενημέρωση στην αποδέσμευση
    - Πρέπει να ψάξουμε για γειτονικές οπές
    - Δεν ξέρουμε όπως σε ποιες λίστες είναι!

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Εικονική μνήμη

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 3:** Διαχείριση Μνήμης

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

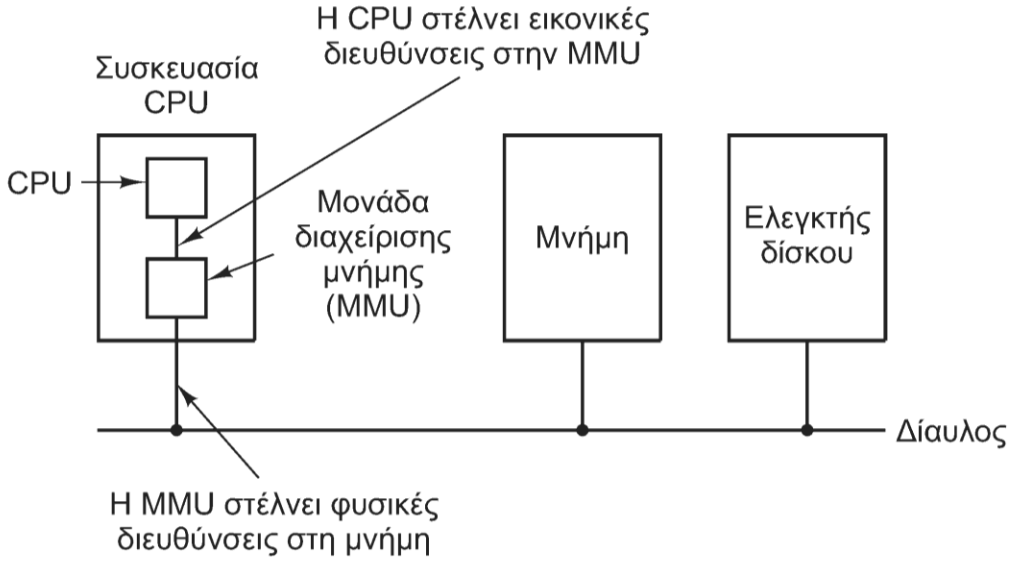
Εικονική μνήμη (1 από 2)

- Γιατί χρειάζεται η εικονική μνήμη;
  - Εκτέλεση τεράστιων προγραμμάτων
  - Εκτέλεση πολλών προγραμμάτων ταυτόχρονα
  - Η εναλλαγή είναι πολύ ακριβή λύση
- Υπερθέματα (overlays)
  - Επιτρέπει εκτέλεση μεγάλων προγραμμάτων
  - Κάθε πρόγραμμα χωρίζεται σε υπερθέματα
    - Αρχικά φορτώνεται ο διαχειριστής υπερθεμάτων
    - Ο διαχειριστής φορτώνει το αρχικό υπέρθεμα
    - Τα άλλα υπερθέματα φορτώνονται όταν χρειάζονται

Εικονική μνήμη (2 από 2)

- Εικονική μνήμη (virtual memory)
  - Κάθε πρόγραμμα έχει δικό του χώρο διευθύνσεων
  - Ο χώρος διαιρείται σε σελίδες σταθερού μεγέθους
  - Ορισμένες σελίδες βρίσκονται στη μνήμη
    - Οι υπόλοιπες βρίσκονται στο δίσκο
  - Το υλικό μεταφράζει αυτόματα τις αναφορές μνήμης
    - Χρησιμοποιεί έναν πίνακα μετάφρασης
  - Οι αναφορές σε μη διαθέσιμες σελίδες παγιδεύονται
    - Το λειτουργικό φορτώνει την αντίστοιχη σελίδα στη μνήμη

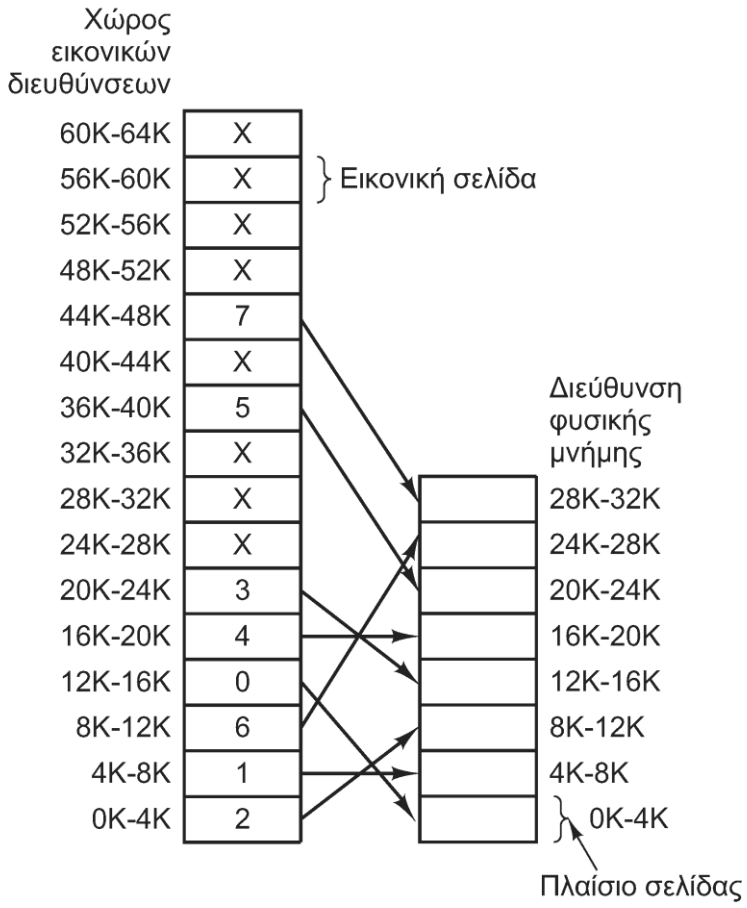
Σελιδοποίηση (1 από 5)



- Σελιδοποίηση (paging)
  - Κάθε πρόγραμμα έχει έναν εικονικό χώρο διευθύνσεων
  - Οι εντολές του αναφέρονται σε εικονικές διευθύνσεις
  - Η μονάδα διαχείρισης μνήμης τις μετατρέπει σε φυσικές
  - Οι φυσικές διευθύνσεις τοποθετούνται στο δίαυλο

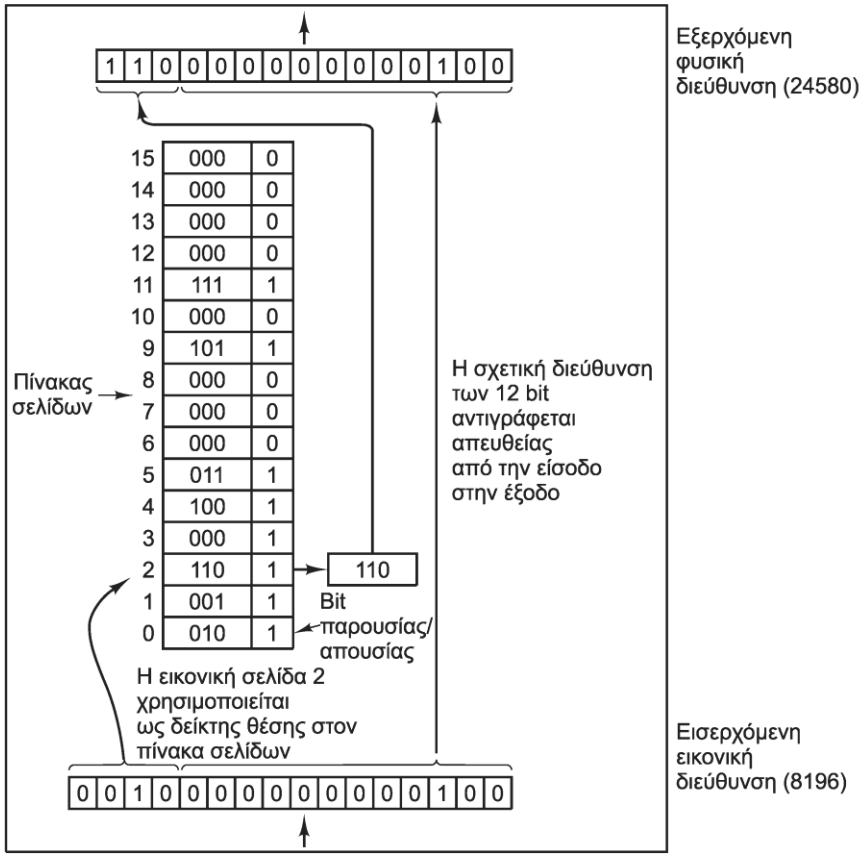
Σελιδοποίηση (2 από 5)

- Παράδειγμα
  - Εικονικός χώρος 64 Κ
    - Διαιρείται σε σελίδες
  - Φυσική μνήμη 32 Κ
    - Διαιρείται σε πλαίσια
  - Σελίδες - πλαίσια 4 Κ
    - 16 σελίδες - 8 πλαίσια
  - Ορισμένες λείπουν (Χ)
    - Σφάλμα αν προσπελαστούν
    - Το ΛΣ ελευθερώνει πλαίσιο και φέρνει τη σελίδα



Σελιδοποίηση (3 από 5)

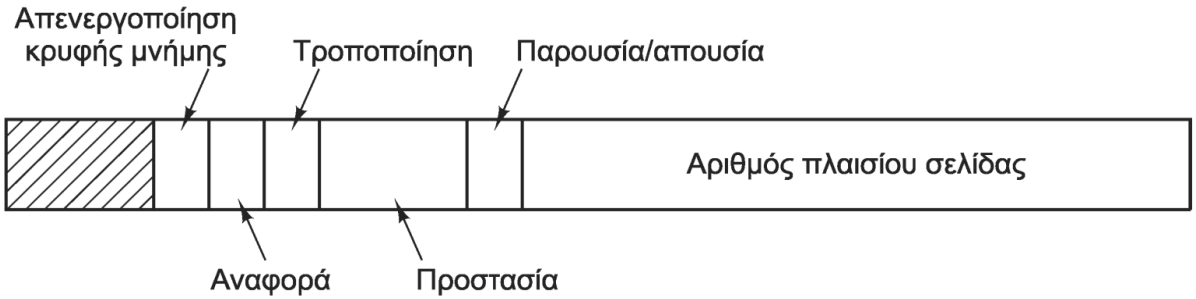
- Λειτουργία MMU
  - Σπάσιμο διεύθυνσης
    - 4 bit σελίδα+12 bit απόσταση
  - Πίνακας χαρτογράφησης
    - Έχει 2^4=16 θέσεις
    - Περιέχει bit παρουσίας
    - Δίνει 3 bit για το πλαίσιο
  - Έλεγχος bit παρουσίας
    - Αν είναι 0, σφάλμα σελίδας
    - Αν είναι 1, διαβάζουμε 3 bit



Σελιδοποίηση (4 από 5)

- Πίνακες σελίδων
  - Κάθε διεύθυνση αποτελείται από 2 μέρη
    - Αριθμός εικονικής σελίδας: περισσότερο σημαντικά bit
    - Απόσταση μέσα στη σελίδα: λιγότερο σημαντικά bit
  - Η διαίρεση καθορίζει το μέγεθος και πλήθος των σελίδων
  - Ο αριθμός εικονικής σελίδας χρησιμοποιείται ως δείκτης
    - Επιλέγει μία καταχώριση από τον πίνακα σελίδων
    - Αν υπάρχει η σελίδα στη μνήμη, διαβάζουμε αριθμό πλαισίου
    - Ο αριθμός πλαισίου επισυνάπτεται στην απόσταση
  - Το αποτέλεσμα στέλνεται στο δίαυλο της μνήμης

Σελιδοποίηση (5 από 5)



- Δομή καταχώρισης πίνακα σελίδων
  - Η ακριβής μορφή εξαρτάται από τη μηχανή
  - Bit προστασίας: τι είδους πρόσβαση επιτρέπεται
    - Μπορεί να είναι RO/RW ή R/W/X
  - Bit τροποποίησης: η σελίδα έχει τροποποιηθεί πρόσφατα
    - Πρέπει να γραφτεί στη μνήμη πριν απομακρυνθεί
  - Bit αναφοράς: η σελίδα έχει χρησιμοποιηθεί πρόσφατα
  - Bit κρυφής μνήμης: όχι αποθήκευση στην κρυφή μνήμη

Επιτάχυνση σελιδοποίησης (1 από 4)

- Η χαρτογράφηση πρέπει να είναι πολύ γρήγορη
  - Κάθε αναφορά στη μνήμη απαιτεί χαρτογράφηση
  - Μία εντολή μπορεί να περιέχει πολλές αναφορές
- Ο πίνακας σελίδων μπορεί να είναι πολύ μεγάλος
  - Εξαρτάται από μέγεθος μνήμης και σελίδας
- Πού βρίσκεται ο πίνακας σελίδων;
  - Σε καταχωρητές: γρήγορη χαρτογράφηση
    - Μεγάλος χρόνος μεταγωγής και πολύ μεγάλο κόστος
  - Στη μνήμη: γρήγορη μεταγωγή (αλλαγή καταχωρητή)
    - Πολύ χαμηλή επίδοση λόγω αναφορών στη μνήμη

Επιτάχυνση σελιδοποίησης (2 από 4)

Έγκυρη	Εικονική σελίδα	Τροποποιημένη	Προστασία	Πλαίσιο σελίδας
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

- Κρυφή μνήμη αναζήτησης μετάφρασης (TLB)
  - Όλοι οι πίνακες σελίδων αρχικά βρίσκονται στη μνήμη
    - Είναι ανέφικτο να έχουμε όλο τον πίνακα σε καταχωρητές
    - Τα προγράμματα χρησιμοποιούν λίγες σελίδες σε κάθε περίοδο
  - Αποθήκευση των μεταφράσεων σε κρυφή μνήμη
    - Ονομάζεται και συνειρμική (associative) μνήμη
    - Αποτελείται από λίγους καταχωρητές μέσα στην MMU

Επιτάχυνση σελιδοποίησης (3 από 4)

- Έστω ότι μια εικονική διεύθυνση στέλνεται στην MMU
  - Η MMU συγκρίνει τον αριθμό σελίδας με την TLB
    - Οι συγκρίσεις γίνονται παράλληλα με όλους τους καταχωρητές
    - Προφανώς αυτό ανεβάζει σημαντικά το κόστος της TLB
  - Αν η σελίδα βρεθεί σχηματίζεται η φυσική διεύθυνση
    - Εκτός αν υπάρχει σφάλμα προστασίας
  - Αν η σελίδα δεν βρεθεί καταφεύγουμε σε πίνακα σελίδων
    - Αντικαθιστά μια από τις υπάρχουσες καταχωρίσεις
  - Όλη η διαδικασία γίνεται μέσω του υλικού της MMU
    - Σε ορισμένα συστήματα (RISC) εμπλέκεται και το λογισμικό

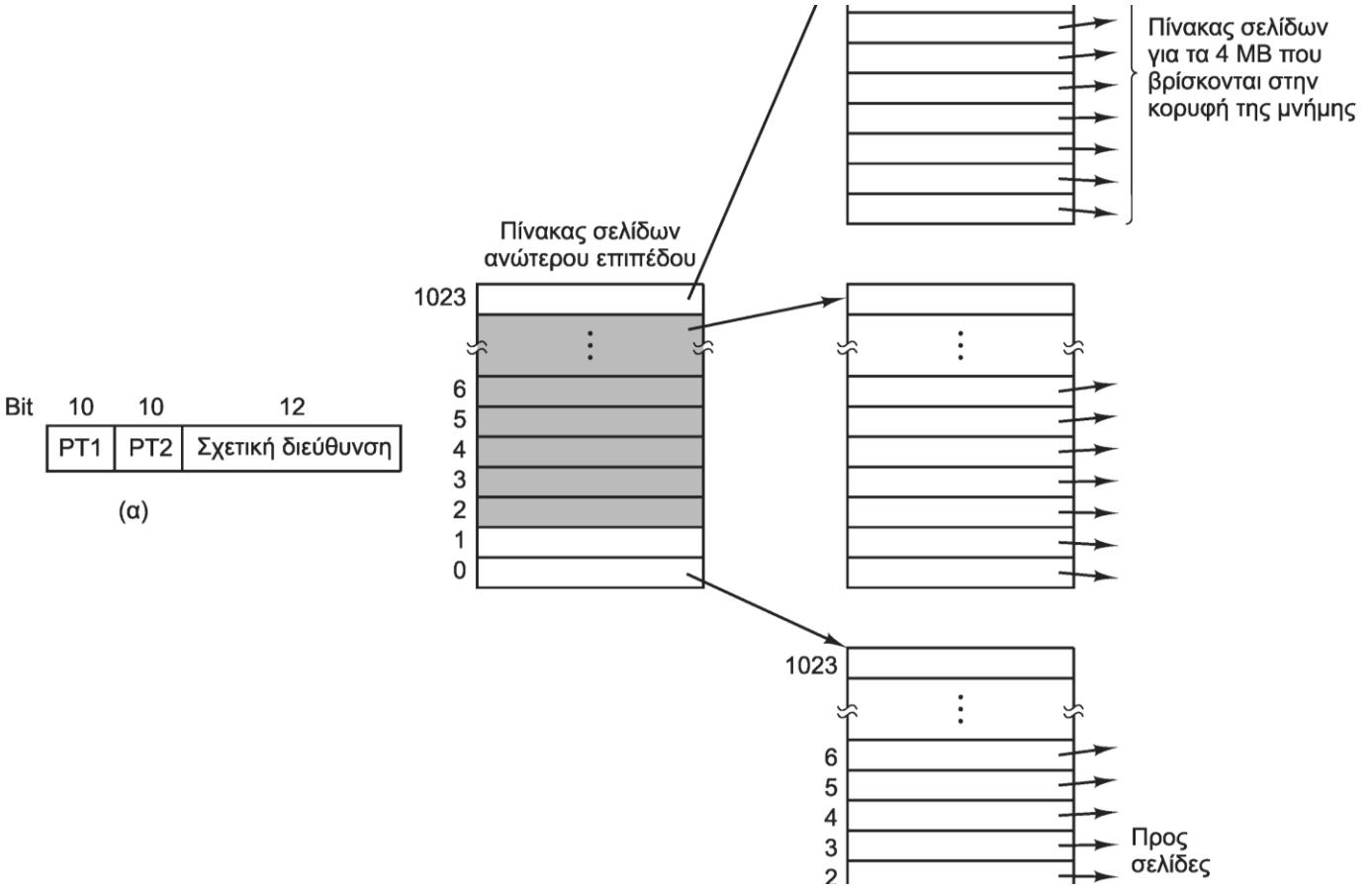
Επιτάχυνση σελιδοποίησης (4 από 4)

- Διαχείριση αστοχιών TLB με λογισμικό
  - Το λειτουργικό σύστημα αντιμετωπίζει την αστοχία
    - Διαβάζει πίνακα σελίδων και δημιουργεί νέα καταχώριση
    - Επιλέγει την καταχώριση που θα αντικατασταθεί
  - Σημαντική απλοποίηση της MMU
  - Επιτρέπει στο λειτουργικό ορισμένες βελτιστοποιήσεις
    - Φόρτωση καταχωρίσεων πριν χρειαστούν οι σελίδες
  - Δύο είδη αστοχιών σελίδων στη διαχείριση με λογισμικό
    - Ήπια αστοχία: η σελίδα είναι στη μνήμη αλλά όχι στην TLB
    - Αυστηρή αστοχία: η σελίδα δεν υπάρχει ούτε στη μνήμη

Πολύ μεγάλες μνήμες (1 από 4)

- Πίνακες σελίδων για μεγάλες μνήμες
  - Παράδειγμα: εικονικές διευθύνσεις 32 bit και σελίδες 4 K
  - Ο πίνακας σελίδων έχει 1 εκατομμύριο καταχωρίσεις!
- Πολυεπίπεδοι πίνακες σελίδων
  - Παράδειγμα: εικονικές διευθύνσεις 32 bit και σελίδες 4 K
    - Τα πρώτα 10 bit αναφέρονται στον πίνακα πρώτου επιπέδου
    - Ο πίνακας χωράει ακριβώς σε μία σελίδα των 4 K
    - Τα επόμενα 10 bit αναφέρονται στον δεύτερο πίνακα
    - Και αυτός ο πίνακας χωράει σε μία σελίδα των 4 K
    - Η απόσταση επισυνάπτεται για να βγει η τελική διεύθυνση

Πολύ μεγάλες μνήμες (2 από 4)

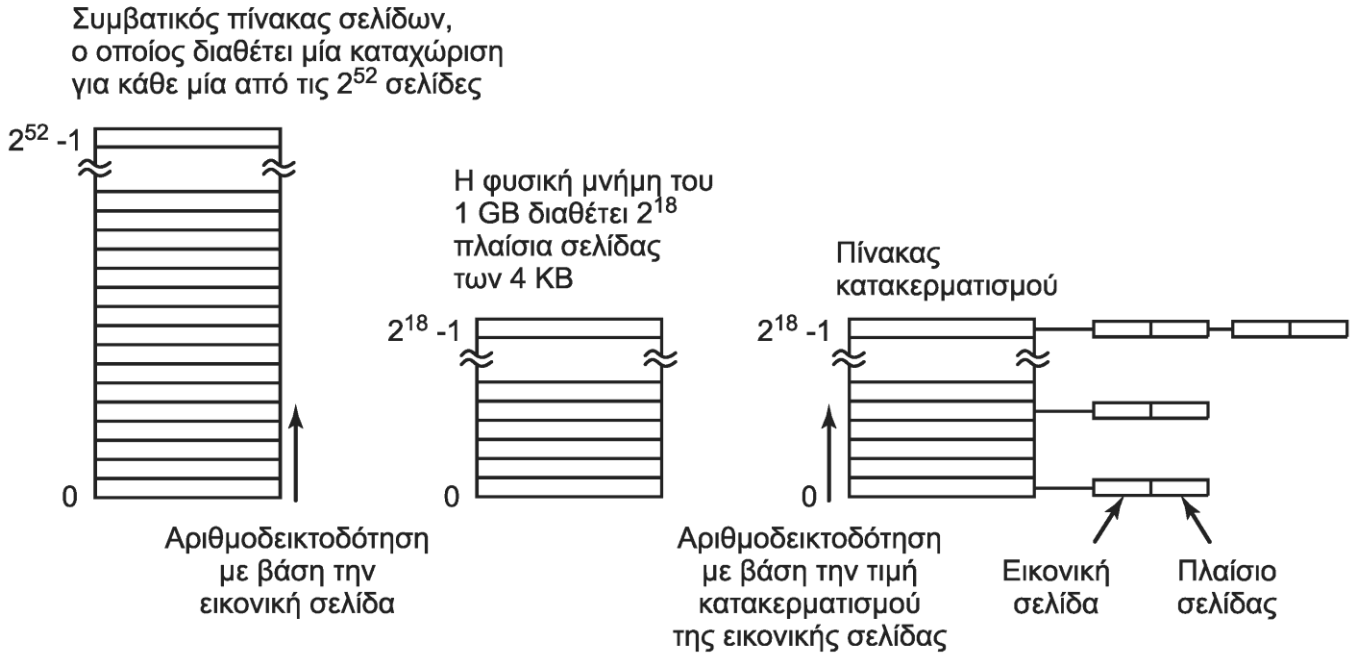


Παράδειγμα πίνακα δύο επιπέδων

Πολύ μεγάλες μνήμες (3 από 4)

- Πολυεπίπεδοι πίνακες σελίδων
  - Τα περισσότερα προγράμματα θέλουν λίγη μνήμη
  - Αρκούν λίγοι πίνακες σελίδων στη μνήμη
    - Πίνακες που δείχνουν σε στοίβα, σωρό και κώδικα
- Ανεστραμμένοι πίνακες σελίδων
  - Οι πολυεπίπεδοι δεν φτάνουν για διευθύνσεις 64 bit
  - Αντί για πίνακα σελίδων, έχουμε πίνακα πλαισίων
    - Δείχνει ποια διεργασία / σελίδα είναι σε κάθε πλαίσιο
    - Ανάλογος με το μέγεθος της φυσικής μνήμης
    - Ανεξάρτητος από το πλήθος των διεργασιών

Πολύ μεγάλες μνήμες (4 από 4)



- Ανεστραμμένοι πίνακες σελίδων
  - Χρήση TLB για τις συχνά χρησιμοποιούμενες σελίδες
  - Χρήση πίνακα κατακερματισμού
    - Κατακερματίζουμε την εικονική διεύθυνση

Αλγόριθμοι αντικατάστασης σελίδων

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 3:** Διαχείριση Μνήμης

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Αλγόριθμοι αντικατάστασης

- Αντικατάσταση σελίδας στη μνήμη
  - Απαιτείται μετά από σφάλμα σελίδας
  - Επιλέγεται μία σελίδα προς απομάκρυνση από τη μνήμη
  - Αν έχει τροποποιηθεί, γράφεται στο δίσκο
  - Στη συνέχεια φορτώνεται στη θέση της η νέα σελίδα
- Αλγόριθμοι αντικατάστασης σελίδων
  - Στόχος: απομάκρυνση μιας «αχρησιμοποίητης» σελίδας
  - Παρόμοιοι αλγόριθμοι με άλλα είδη αντικατάστασης
  - Η χρονική κλίμακα είναι σχετικά μεγάλη
    - Η νέα σελίδα πρέπει να φορτωθεί από το δίσκο

Ο βέλτιστος αλγόριθμος

- Βέλτιστος αλγόριθμος αντικατάστασης (OPT)
  - Επιλέγεται η σελίδα που θα χρησιμοποιηθεί πιο αργά
  - Ο αλγόριθμος είναι ο θεωρητικά βέλτιστος
  - Δεν ξέρουμε πότε θα ξαναχρησιμοποιηθεί κάθε σελίδα
  - Μπορούμε να τον δοκιμάσουμε με ένα πρόγραμμα
    - Σημειώνουμε όλες τις αναφορές σε σελίδες
    - Προσομοιώνουμε τον βέλτιστο αλγόριθμο
    - Χρησιμοποιούμε τα αποτελέσματα ως μέτρο σύγκρισης
    - Ένας αλγόριθμος 1% χειρότερος είναι αρκετά καλός
    - Τα αποτελέσματα ισχύουν για τη συγκεκριμένη εκτέλεση

Αλγόριθμος NRU (1 από 3)

- Αλγόριθμος NRU
  - Αξιοποίηση bit κατάστασης A και T
    - Συνήθως το υλικό ενημερώνει αυτόματα αυτά τα bit
  - Προσομοίωση με λογισμικό
    - Αρχικά όλες οι σελίδες σημειώνονται ως απούσες
    - Όταν γίνει αναφορά σε σελίδα έχουμε σφάλμα σελίδας
    - Το λειτουργικό σύστημα θέτει το bit A
    - Η σελίδα σημειώνεται ως ανάγνωσης μόνο
    - Όταν γίνει εγγραφή στη σελίδα έχουμε σφάλμα σελίδας
    - Το λειτουργικό σύστημα θέτει το bit T
    - Η σελίδα σημειώνεται ως ανάγνωσης/εγγραφής

Αλγόριθμος NRU (2 από 3)

- Αλγόριθμος NRU
  - Κατανομή σελίδων σε 4 κατηγορίες
    - 0: δεν έγινε αναφορά, δεν τροποποιήθηκε
    - 1: δεν έγινε αναφορά, τροποποιήθηκε
    - 2: έγινε αναφορά, δεν τροποποιήθηκε
    - 4: έγινε αναφορά, τροποποιήθηκε
  - Ο NRU επιλέγει σελίδα από τη χαμηλότερη
    - Προτεραιότητα σε παλιές σελίδες
    - Δευτερευόντως σε «καθαρές» σελίδες

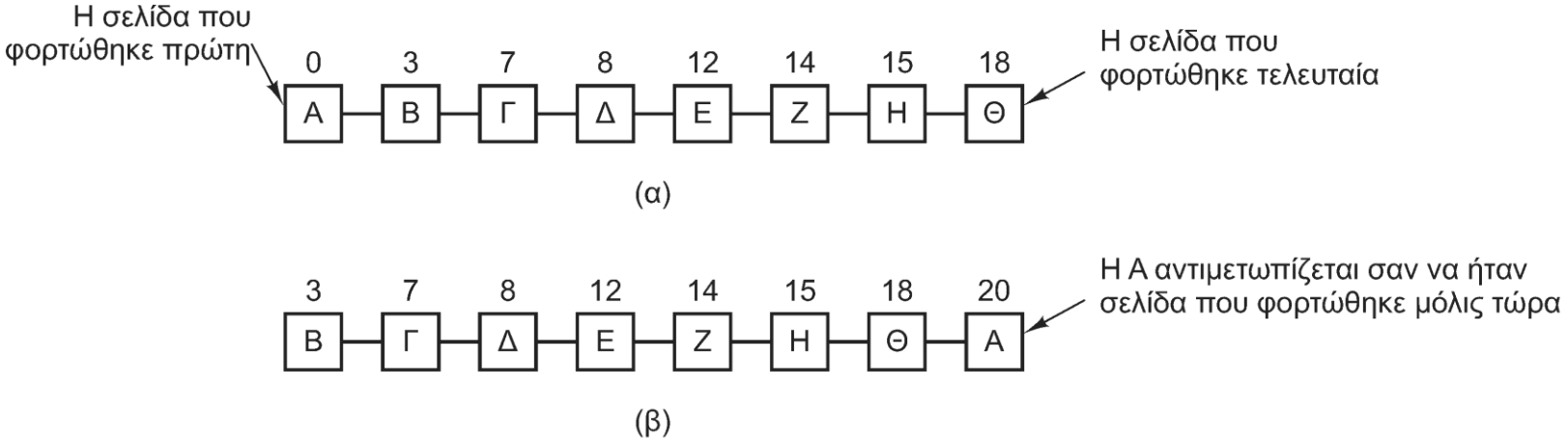
Αλγόριθμος NRU (3 από 3)

- Αλγόριθμος NRU
  - Πώς μπορεί να υπάρχει η κατηγορία 1;
    - 1: δεν έγινε αναφορά, τροποποιήθηκε
  - Το λειτουργικό περιοδικά μηδενίζει τα bit A
    - Μας ενδιαφέρουν μόνο οι πρόσφατες αναφορές
  - Τα bit T μηδενίζονται αν γραφτεί η σελίδα
    - Αυτό μπορεί να γίνει στο παρασκήνιο
    - Μία σελίδα μπορεί έτσι να γίνει καθαρή

Αλγόριθμος FIFO

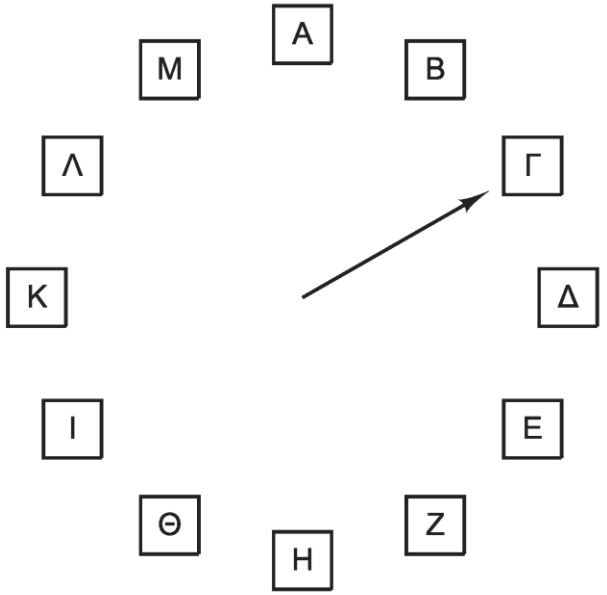
- Αλγόριθμος FIFO
  - Ταξινόμηση σελίδων με τη σειρά φόρτωσης
    - Αρκεί μια γραμμική λίστα των σελίδων
    - Κάθε νέα σελίδα μπαίνει στο τέλος
    - Η παλιότερη σελίδα βρίσκεται στην αρχή
  - Επιλέγεται η σελίδα που φορτώθηκε πρώτη
    - Η σελίδα που είναι περισσότερη στη μνήμη
    - Η σελίδα αυτή μπορεί να χρησιμοποιείται πολύ!

Αλγόριθμος δεύτερης ευκαιρίας



- Αλγόριθμος δεύτερης ευκαιρίας
  - Αρχικά επιλέγει την παλιότερη σελίδα (όπως ο FIFO)
  - Αν το bit A είναι 1, μηδενίζουμε και την βάζουμε στο τέλος
  - Αλλιώς η σελίδα επιλέγεται για αντικατάσταση
  - Ψάχνει για μια παλιά και αχρησιμοποίητη σελίδα
  - Αν όλες έχουν χρησιμοποιηθεί, εκφυλίζεται σε FIFO

Αλγόριθμος του ρολογιού



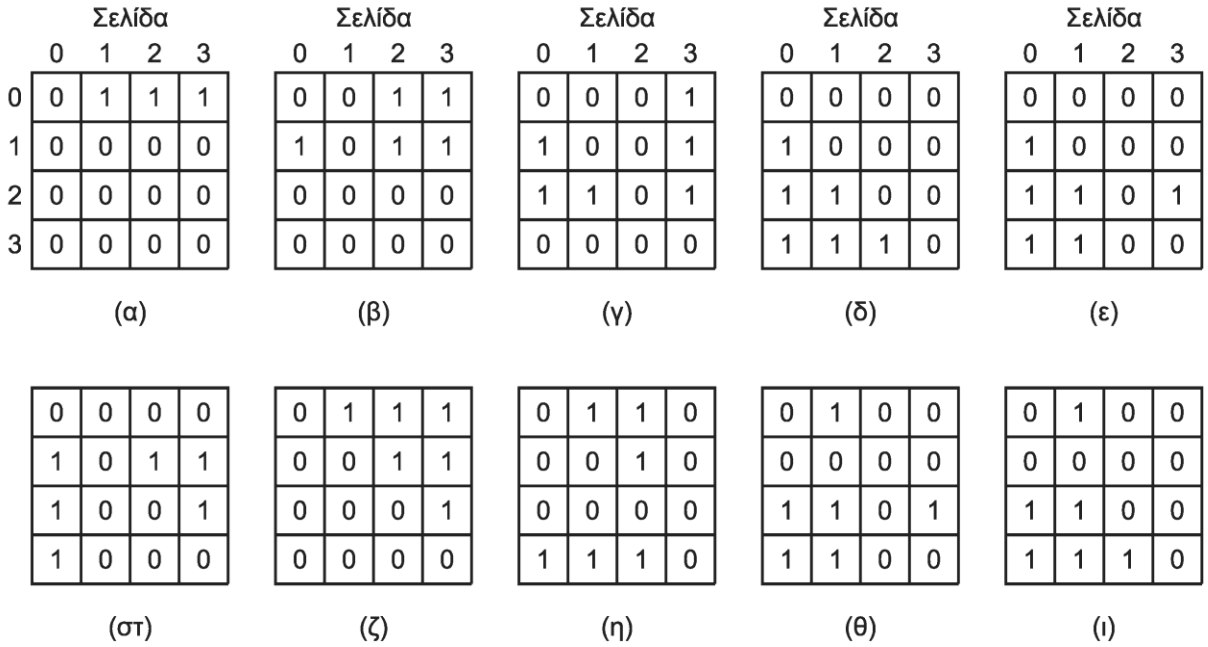
Όταν συμβαίνει σφάλμα σελίδας, τότε εξετάζεται η σελίδα στην οποία δείχνει ο δείκτης.
Η επόμενη ενέργεια εξαρτάται από το bit A:
A = 0: Αφαιρείται η σελίδα
A = 1: Το A γίνεται 0 και ο δείκτης προχωρεί κατά μία θέση

- Αλγόριθμος του ρολογιού
  - Δεύτερη ευκαιρία με κυκλική λίστα
  - Σε κάθε σφάλμα ελέγχουμε τη σελίδα του δείκτη
    - Αν έχει χρησιμοποιηθεί, μηδενίζουμε το bit A και προχωράμε
    - Αν όχι, η σελίδα επιλέγεται για αντικατάσταση

Αλγόριθμος LRU (1 από 2)

- Αλγόριθμος LRU
  - Επιλογή της σελίδας που προσπελάστηκε παλιότερα
  - Η ακριβής υλοποίηση έχει απαγορευτικό κόστος
    - Απαιτεί συνεχή ενημέρωση μιας λίστας
- Προσεγγιστικές υλοποιήσεις του LRU
  - Το υλικό έχει έναν μετρητή των 64 bit
  - Ο μετρητής αυξάνεται κατά 1 μετά από κάθε εντολή
  - Σε κάθε αναφορά στη μνήμη ο μετρητής αποθηκεύεται
    - Στην καταχώριση στον πίνακα σελίδων
    - Για λογικό κόστος, θα πρέπει να γίνεται στο TLB

Αλγόριθμος LRU (2 από 2)

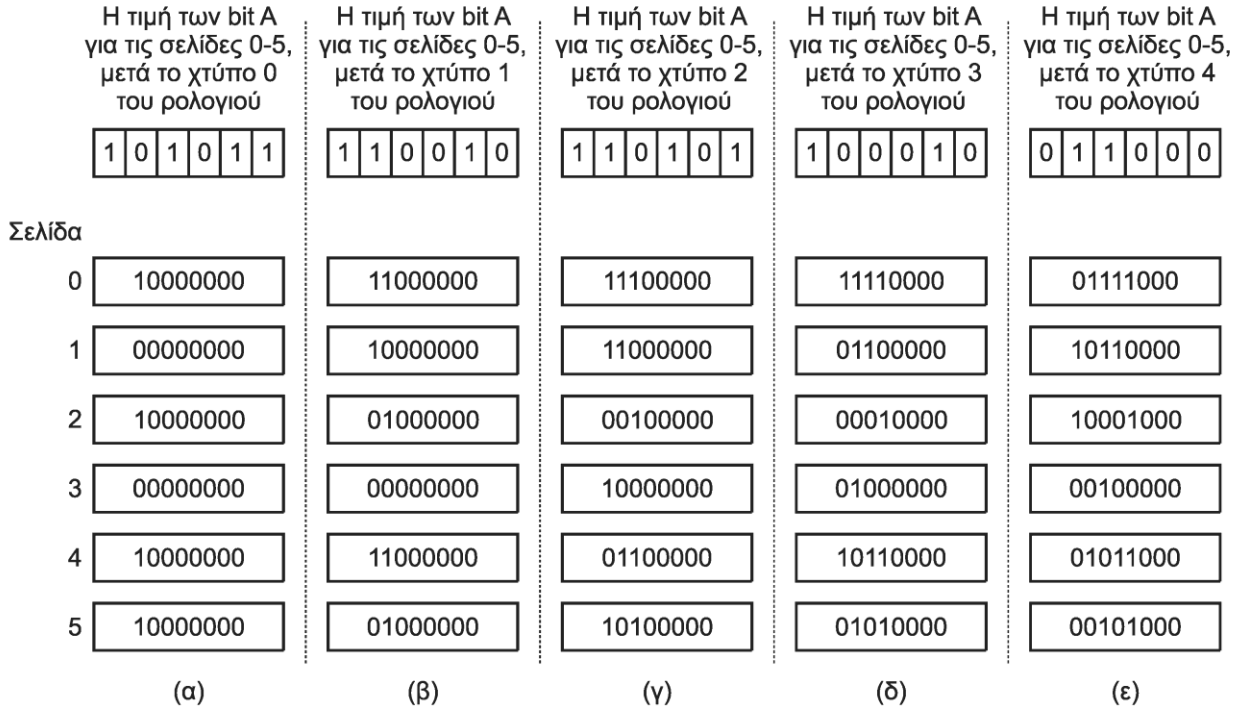


- Προσεγγιστικές υλοποιήσεις του LRU
  - Για n σελίδες έχουμε πίνακα n x n bit, αρχικά μηδενισμένο
  - Αναφορά στην k: γραμμή k γίνεται 1 και στήλη k γίνεται 0
  - Η στήλη με τα πιο πολλά 1 χρησιμοποιήθηκε πιο παλιά

Προσομοίωση LRU (1 από 3)

- Προσομοίωση του αλγόριθμου LRU με λογισμικό
  - Οι υπολογιστές δεν διαθέτουν υλικό για τον LRU
- Αλγόριθμος NFU
  - Κάθε σελίδα έχει έναν μετρητή, αρχικά μηδενισμένο
  - Σε κάθε διακοπή του ρολογιού σαρώνονται οι σελίδες
    - Το bit A προστίθεται στον μετρητή και μηδενίζεται
  - Αρκετά καλή προσέγγιση του LRU
  - Το πρόβλημα είναι ότι ο NFU δεν ξεχνάει τίποτα!
    - Οι μετρητές δεν μειώνονται με το χρόνο

Προσομοίωση LRU (2 από 3)



- Γήρανση (aging) των μετρητών
  - Οι μετρητές ολισθαίνουν δεξιά πριν από κάθε πρόσθεση
  - Το bit A προστίθεται στο αριστερό και όχι στο δεξί άκρο

Προσομοίωση LRU (3 από 3)

- Αλγόριθμος NFU
  - Με την γήρανση οι μετρητές σταδιακά μηδενίζονται
  - Η τελευταία τιμή έχει πολύ μεγαλύτερο βάρος
- Αποκλίσεις NFU από LRU
  - Οι μετρητές είναι προσεγγιστικοί
    - Έστω δύο σελίδες που χρησιμοποιήθηκαν στο ίδιο διάστημα
    - Δεν φαίνεται ποια χρησιμοποιήθηκε πιο πρόσφατα
  - Οι μετρητές έχουν περιορισμένη μνήμη
    - Μετά από λίγη ώρα μηδενίζονται όλοι
  - Συνήθως αυτά δεν έχουν μεγάλη σημασία

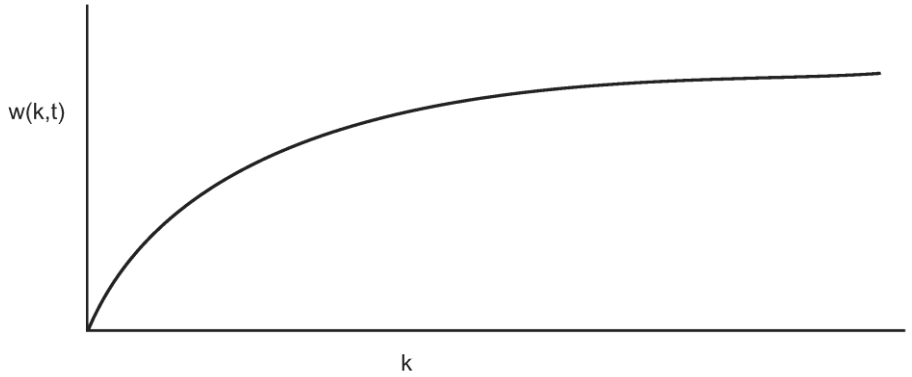
Σύνολο εργασίας (1 από 4)

- Σελιδοποίηση κατ'απαίτηση
  - Όταν ξεκινά η διεργασία, δεν έχει καμία σελίδα στη μνήμη
  - Οι σελίδες φορτώνονται όταν συμβεί σφάλμα σελίδας
- Σύνολο εργασίας διεργασίας
  - Οι σελίδες που χρησιμοποιεί την τρέχουσα περίοδο
  - Περιορισμένο λόγω της τοπικότητας των αναφορών
    - Σε κάθε φάση της μια διεργασία χρησιμοποιεί λίγες σελίδες
  - Αρκεί το σύνολο εργασίας να είναι στη μνήμη
  - Αλλιώς θα έχει συνεχώς σφάλματα σελίδες
    - Σε αυτή την περίπτωση λέμε ότι η διαδικασία «αλωνίζει»

Σύνολο εργασίας (2 από 4)

- Οι διεργασίες πρέπει ενίοτε να πηγαίνουν στο δίσκο
  - Ώστε να υπάρχει αρκετός χώρος για άλλες διεργασίες
- Τι θα γίνει όταν επανέλθει μια διεργασία στη μνήμη;
  - Η απλούστερη λύση είναι να μην γίνει τίποτα
    - Η διεργασία σταδιακά θα φορτώσει το σύνολο εργασίας της
    - Αυτό όμως θα έχει ως κόστος πολλά σφάλματα σελίδων
  - Σε μερικά ΛΣ παρακολουθείται το σύνολο εργασίας
  - Προσελιδοποίηση: φόρτωση του συνόλου εργασίας
    - Όταν επανέρχεται στη μνήμη η διεργασία
    - Αποφυγή των υπερβολικών σφαλμάτων σελίδων

Σύνολο εργασίας (3 από 4)

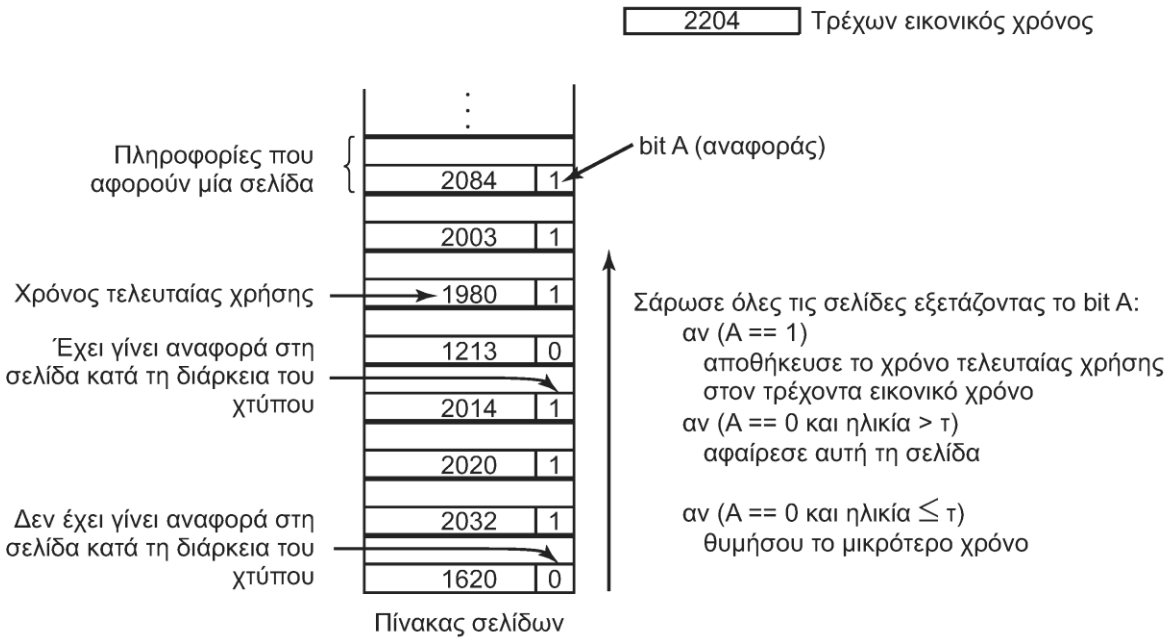


- Μοντελοποίηση συνόλου εργασίας
  - Ορίζουμε το σύνολο εργασίας τη στιγμή t ως w(k,t)
    - Περιλαμβάνει τις σελίδες των τελευταίων k αναφορών
    - Μη φθίνουσα συνάρτηση του k, τείνει προς μέγεθος διεργασίας
    - Μικρότερο αν δεν χρησιμοποιούνται όλες οι σελίδες
  - Η καμπύλη αυξάνεται απότομα και μετά ομαλοποιείται
    - Οι διεργασίες αναφέρονται σε λίγες σελίδες κάθε στιγμή
    - Οι σελίδες αλλάζουν αργά με την πρόοδο του προγράμματος

Σύνολο εργασίας (4 από 4)

- Συμπέρασμα: το σύνολο εργασίας βοηθάει
  - Προσελιδοποίηση σελίδων που είναι στο σύνολο εργασίας
  - Επιλογή σελίδων εκτός συνόλου για αντικατάσταση
- Παρακολούθηση συνόλου εργασίας
  - Παρακολούθηση των τελευταίων k αναφορών
  - Η ακριβής παρακολούθηση έχει απαγορευτικό κόστος
  - Συνήθως χρησιμοποιούμε κάποια προσέγγιση
- Παράδειγμα: αναφορές στα τελευταία 100 ms
  - Μιλάμε πάντα για εικονικό και όχι πραγματικό χρόνο
    - Δηλαδή τον χρόνο πραγματικής εκτέλεσης της διεργασίας

Αλγόριθμος συνόλου εργασίας (1 από 2)



- Αλγόριθμος αντικατάστασης συνόλου εργασίας
  - Δύο πεδία: bit A και τελευταία προσπέλαση
    - Αν A == 1 σημειώνουμε τον τρέχοντα εικονικό χρόνο
    - Αν A == 0 τότε δεν έχει γίνει αναφορά στο τελευταίο διάστημα

Αλγόριθμος συνόλου εργασίας (2 από 2)

- Αλγόριθμος αντικατάστασης συνόλου εργασίας
  - Υπολογίζουμε την ηλικία της σελίδας
    - Τρέχων χρόνος μείον χρόνος τελευταίας προσπέλασης
  - Αν η ηλικία είναι πάνω από t τότε αντικαθίσταται
    - Συνεχίζουμε μέχρι να αντικαταστήσουμε όλες τις σελίδες
  - Αν η ηλικία είναι κάτω από t τότε
    - Σημειώνουμε την παλιότερη τέτοια σελίδα
    - Αν δεν αντικατασταθεί άλλη, τότε αντικαθίσταται αυτή

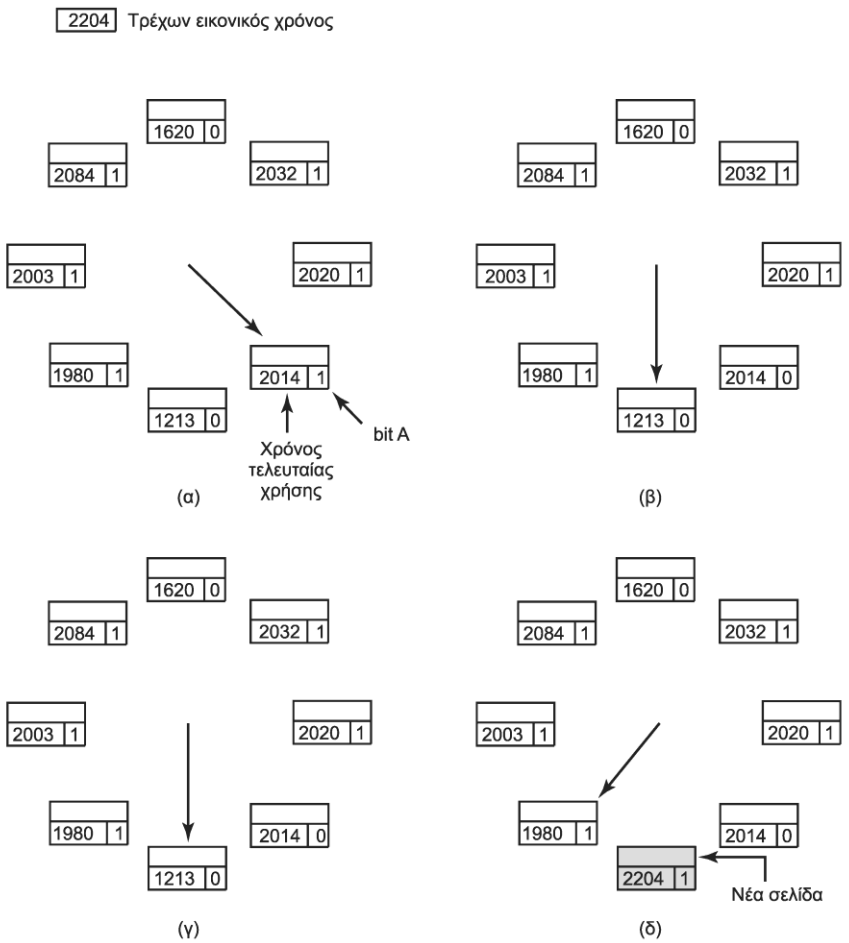
Αλγόριθμος WSClock (1 από 2)

- Αλγόριθμος WSClock
  - Οι σελίδες οργανώνονται σε κυκλική λίστα
  - Σε κάθε σφάλμα ξεκινάμε από εκεί που είχαμε μείνει
  - Αν η σελίδα έχει A == 1 μηδενίζουμε και προχωράμε
  - Αν έχει A == 0 τότε συγκρίνουμε την ηλικία με το t
    - Αν είναι μεγαλύτερη από t τότε η σελίδα επιλέγεται
  - Αν είναι καθαρή, επιλέγεται και αντικαθίσταται
    - Αν δεν είναι καθαρή, προγραμματίζεται για εγγραφή
    - Προχωράμε στην επόμενη αφού η εγγραφή θα αργήσει

Αλγόριθμος WSClock (2 από 2)

- Αλγόριθμος WSClock
  - Αν σταλθούν n σελίδες στο δίσκο, σταματάμε
    - Δεν θέλουμε να καλείται συνέχεια ο αλγόριθμος
    - Αλλά δεν θέλουμε και να αδειάσουμε τη μνήμη
  - Αν δεν έχει βρεθεί σελίδα σε έναν κύκλο
    - Αν έχουν σταλθεί σελίδες στο δίσκο συνεχίζουμε
      - Τελικά κάποια θα αδειάσει και θα αντικατασταθεί
    - Αν δεν έχουν σταλθεί σελίδες στο δίσκο
      - Επιλέγουμε μία καθαρή σελίδα στην τύχη!

Αλγόριθμος WSClock (3)



Παράδειγμα αλγόριθμου WSClock

Σύνοψη αλγορίθμων

Αλγόριθμος	Περιγραφή
Βέλτιστος	Δεν είναι δυνατόν να υλοποιηθεί, αλλά χρησιμοποιείται ως μέτρο σύγκρισης
NRU (Not Recently Used)	Πολύ χονδροειδής
FIFO (First-In, First-Out)	Υπάρχει περίπτωση να αφαιρέσει σημαντικές σελίδες
Δεύτερης ευκαιρίας	Πολύ βελτιωμένος σε σχέση με τον FIFO
Ρολογιού	Ρεαλιστικός
LRU (Least Recently Used)	Εξαιρετικός, αλλά η ακριβής υλοποίησή του είναι δύσκολη
NFU (Not Recently Used)	Μάλλον χονδροειδής προσέγγιση του LRU
Γήρανσης	Αποδοτικός αλγόριθμος, καλή προσέγγιση του LRU
Συνόλου εργασίας	Κάπως ακριβή υλοποίηση
WSClock	Καλός και αποδοτικός αλγόριθμος

- Κόστος υλοποίησης και αποτελεσματικότητα
  - Ορισμένοι απαιτούν ειδικό υλικό (π.χ. LRU)
  - Οι αλγόριθμοι γήρανσης και WSClock είναι πιο πρακτικοί
    - Προσεγγίσεις LRU και συνόλου εργασίας, αντίστοιχα

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

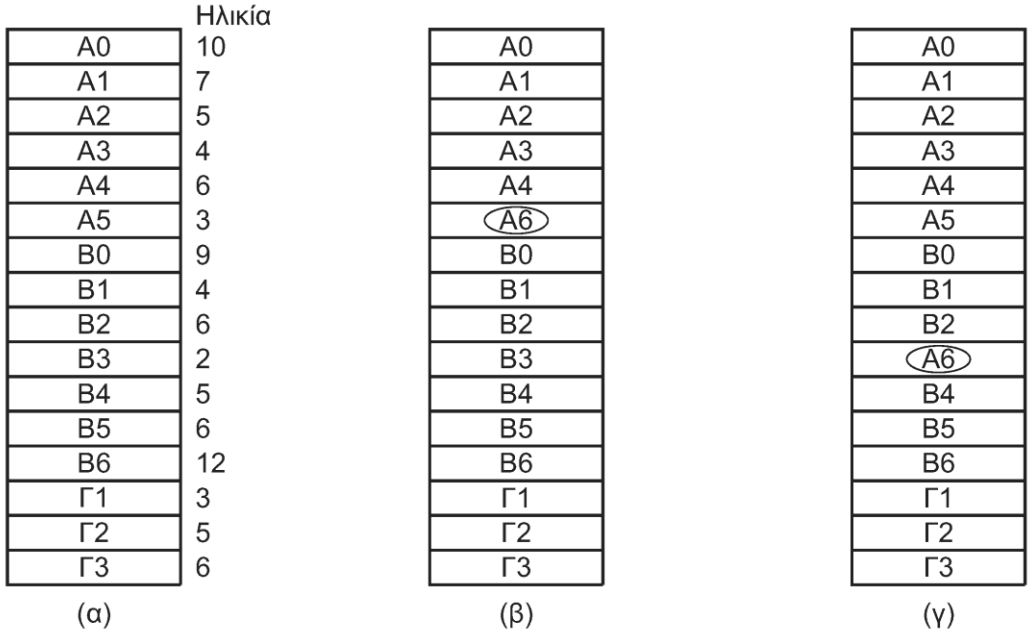
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Θέματα σχεδιασμού

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 3:** Διαχείριση Μνήμης

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Τοπικές και καθολικές πολιτικές (1 από 3)

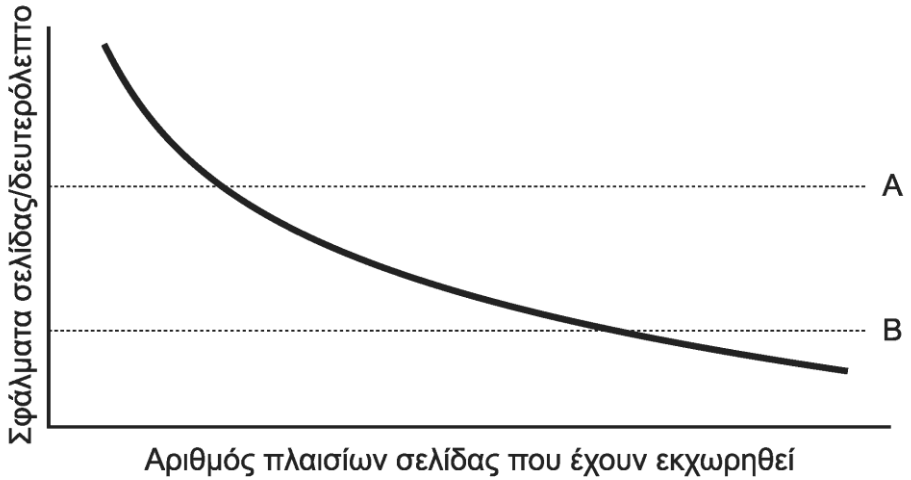


- Πώς πρέπει να κατανέμεται η μνήμη στις διεργασίες;
  - Τοπικά: αντικατάσταση σελίδων της ίδιας διεργασίας
  - Καθολικά: αντικατάσταση οποιονδήποτε σελίδων
  - Η καθολική πολιτική κατανέμει τη μνήμη δυναμικά

Τοπικές και καθολικές πολιτικές (2 από 3)

- Γενικά οι καθολικές πολιτικές λειτουργούν καλύτερα
  - Έστω ότι η διεργασία αλλάζει μέγεθος συνόλου εργασίας
  - Αν μεγαλώσει, η τοπική θα οδηγήσει σε αλώνισμα
  - Αν μικρύνει, η τοπική πολιτική θα σπαταλά μνήμη
- Πώς όμως θα κατανείμουμε τη μνήμη;
  - Καταμερισμός διαθέσιμων πλαισίων στα ίσα
    - Η κατανομή αλλάζει με το πλήθος των διεργασιών
  - Καταμερισμός διαθέσιμων πλαισίων αναλόγως μεγέθους
    - Πρέπει να δίνεται όμως ένας ελάχιστος αριθμός σε όλες
  - Αρχική κατανομή και μετά δυναμική προσαρμογή

Τοπικές και καθολικές πολιτικές (3 από 3)



- Αλγόριθμος Συχνότητας Σφαλμάτων Σελίδας (PFF)
  - Συνήθως το PFF μειώνεται με πρόσθετη μνήμη
  - Παρακολούθηση συχνότητας σφαλμάτων ανά διεργασία
    - Μπορεί να χρησιμοποιεί σταθμισμένους μέσους
  - Αν ανέβει πολύ θέλουμε περισσότερη μνήμη
  - Αν πέσει πολύ μπορούμε να παραχωρήσουμε μνήμη

Έλεγχος φορτίου

- Αν δεν χωράνε όλα τα σύνολα εργασίας;
  - Τότε έχουμε αλώνισμα (συνεχή αλλαγή σελίδων)
  - Ο PFF δείχνει ότι όλες οι διεργασίες θέλουν μνήμη
- Πρέπει να εξαλειφθούν μερικές διεργασίες
  - Μεταφέρονται στο δίσκο οι σελίδες τους
- Χρονοπρογραμματισμός σε δύο επίπεδα
  - Έλεγχος φορτίου με μεταφορά στο δίσκο
  - Καταμερισμός μνήμης με σελιδοποίηση
  - Πρέπει να φροντίζουμε να έχουμε αρκετές στη μνήμη

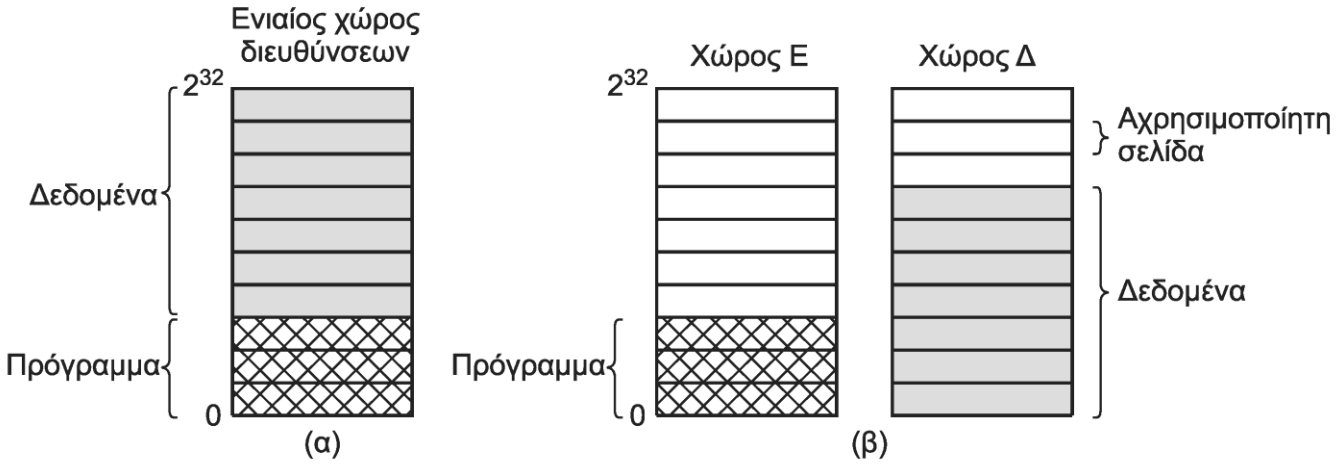
Μέγεθος σελίδας (1 από 2)

- Το υλικό παρέχει κάποιο βασικό μέγεθος σελίδας
  - Το λογισμικό μπορεί να λειτουργεί με πολλαπλάσια
- Το μέγεθος σελίδας αποτελεί συμβιβασμό
  - Οι μεγάλες σελίδες αυξάνουν την εσωτερική κατάτμηση
    - Ένα τμήμα δεν θα χωράει ακριβώς σε σελίδες
    - Η τελευταία σελίδα θα είναι κατά μέσο όρο μισογεμάτη
  - Οι μικρές σελίδες απαιτούν μεγάλο πίνακα σελίδων
    - Περισσότερος χρόνος κατά την αλλαγή διεργασίας
    - Το κόστος μεταφοράς ανά σελίδα είναι σταθερό
    - Συμφέρει να μεταφέρουμε λίγες μεγάλες σελίδες

Μέγεθος σελίδας (2 από 2)

- Μοντελοποίηση της επιβάρυνσης
  - μ byte ανά διεργασία
  - σ byte ανά σελίδα
  - κ byte ανά καταχώριση στον πίνακα σελίδων
  - Κάθε σελίδα απαιτεί μ/σ σελίδες
  - Άρα ο πίνακας σελίδων έχει μέγεθος μκ/σ
  - Η εσωτερική κατάτμηση είναι σ/2
  - Η επιβάρυνση ανά διεργασία είναι μκ/σ+σ/2
  - Η βέλτιστη τιμή του σ είναι ρίζα του 2μκ
  - Παράδειγμα: μ = 1 MB και κ = 8 byte, βέλτιστο σ = 4 KB

Εντολές και δεδομένα

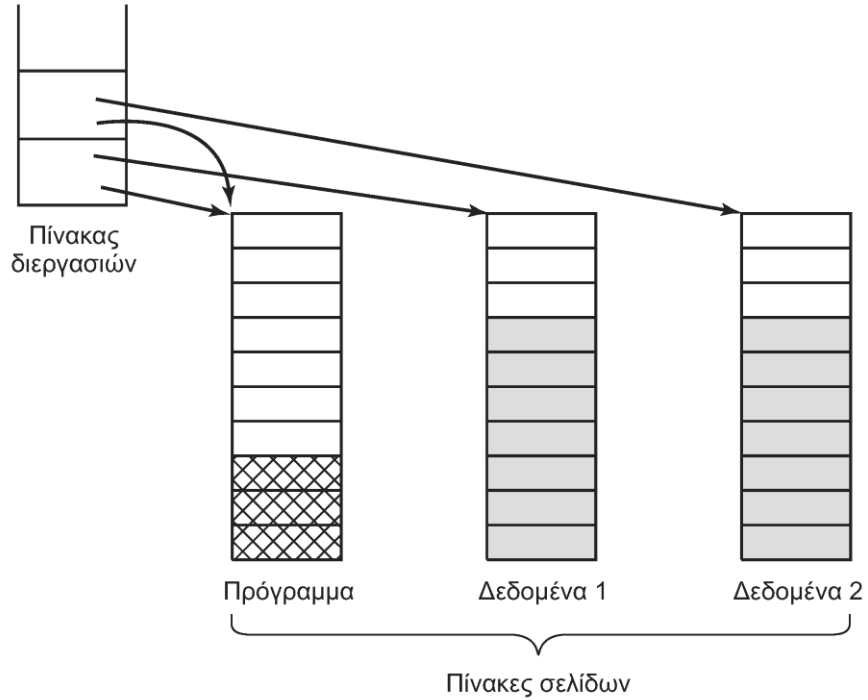


- Διαχωρισμός χώρων εντολών και δεδομένων
  - Συνήθως δεδομένα και εντολές είναι σε ενιαίο χώρο
  - Σε ορισμένα συστήματα διαχωρίζονται
    - Οι εντολές αναφέρονται σε άλλη μνήμη από τα δεδομένα
  - Δύο ανεξάρτητες εικονικές μνήμες
    - Χρησιμοποιήθηκε στο PDP-11 για μεγάλα προγράμματα

Κοινόχρηστες σελίδες (1 από 3)

- Κοινόχρηστες σελίδες
  - Το πρόγραμμα μπορεί να εκτελείται σε πολλές διεργασίες
  - Θα μπορούσαν οι σχετικές σελίδες να καταμερίζονται
    - Επιτρέπεται στις σελίδες ανάγνωσης μόνο
    - Στην πράξη, είναι οι σελίδες κώδικα
  - Υλοποίηση με χωριστούς χώρους διευθύνσεων
    - Πίνακας σελίδων δεδομένων και κώδικα ανά διεργασία
    - Ο πίνακας σελίδων κώδικα μπορεί να καταμερίζεται
  - Λίγο πιο περίπλοκη με ενιαίο χώρο διευθύνσεων
  - Τι γίνεται αν μία από τις διεργασίες φύγει από τη μνήμη;

Κοινόχρηστες σελίδες (2 από 3)



- Παράδειγμα καταμερισμού σελίδων
  - Δύο διεργασίες έχουν τον ίδιο κώδικα
  - Τα δεδομένα όμως διαφοροποιούνται

Κοινόχρηστες σελίδες (3 από 3)

- Αντιγραφή κατά την εγγραφή
  - UNIX: μετά το fork οι διεργασίες μοιράζονται δεδομένα
  - Κάθε διεργασία έχει άλλο πίνακα αλλά τις ίδιες σελίδες
  - Οι σελίδες όμως σημειώνονται ως μόνο ανάγνωσης
  - Όταν τροποποιηθεί μία σελίδα έχουμε παγίδα
  - Το λειτουργικό δημιουργεί αντίγραφο της σελίδας
    - Κάθε διεργασία διαθέτει τώρα το δικό της αντίγραφο
  - Τα αντίγραφα γίνονται ανάγνωσης/εγγραφής
  - Όποιες σελίδες δεν τροποποιηθούν δεν θα αντιγραφούν

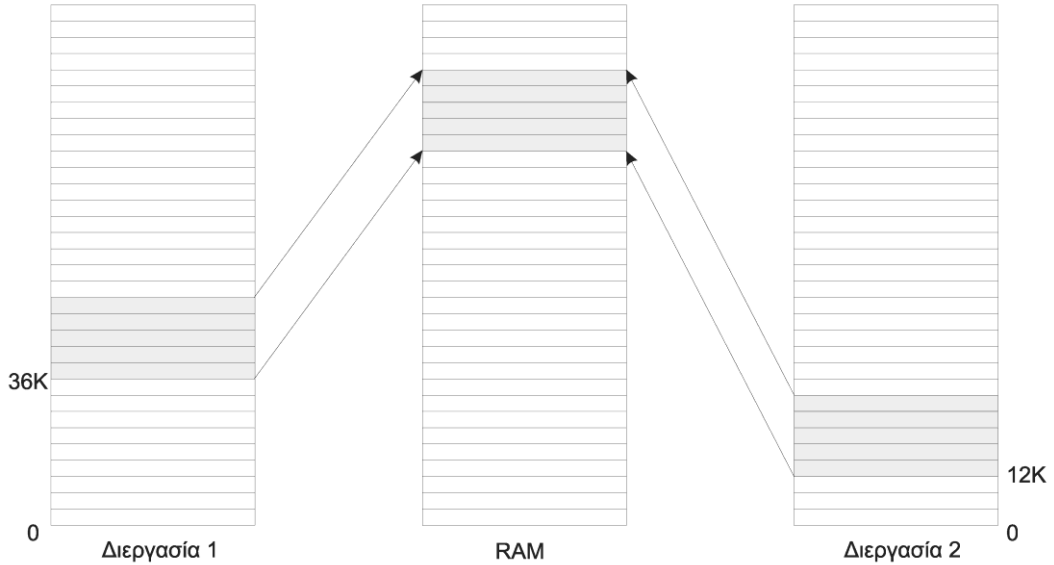
Κοινόχρηστες βιβλιοθήκες (1 από 3)

- Κοινόχρηστες βιβλιοθήκες
  - Οι διεργασίες συχνά χρησιμοποιούν μεγάλες βιβλιοθήκες
    - Για παράδειγμα, τις βιβλιοθήκες γραφικών του GUI
  - Με στατική σύνδεση είναι δύσκολο να τις μοιραστούμε
    - Κάθε διεργασία έχει διαφορετικά τμήμα σε διαφορετικά μέρη
  - Συμφέρει να έχουμε δυναμική σύνδεση
  - Windows: Dynamic Link Libraries (DLL)
    - Το πρόγραμμα συνδέεται με ρουτίνες στελέχους
    - Αυτές δεσμεύονται με τη βιβλιοθήκη στο χρόνο εκτέλεσης

Κοινόχρηστες βιβλιοθήκες (2 από 3)

- Λειτουργία DLL στα Windows
  - Οι DLL φορτώνονται ολόκληρες στη μνήμη
    - Δεν φορτώνονται μόνο οι συναρτήσεις που χρησιμοποιούνται
    - Στην πραγματικότητα σελιδοποιούνται και αυτές
    - Φόρτωση είτε με το πρόγραμμα είτε με την πρώτη κλήση
  - Όλα τα προγράμματα έχουν πρόσβαση στις DLL
  - Τα DLL επιτρέπουν τη διόρθωση σφαλμάτων
    - Έστω ότι βρέθηκε κάποιο πρόγραμμα ασφάλειας σε ένα DLL
    - Μπορεί να αντικατασταθεί με νέα έκδοση
    - Οι διεργασίες βλέπουν τις ίδιες ακριβώς κλήσεις

Κοινόχρηστες βιβλιοθήκες (3 από 3)



- Κοινόχρηστες βιβλιοθήκες
  - Κάθε διεργασία φορτώνει τη βιβλιοθήκη αλλού
  - Η βιβλιοθήκη πρέπει να είναι ανεξάρτητη θέσης
    - Να εκτελείται οπουδήποτε στη μνήμη
    - Άρα να χρησιμοποιεί μόνο σχετικά άλματα και αναφορές

Χαρτογραφημένα αρχεία

- Χαρτογραφημένα αρχεία στη μνήμη
  - Παρέχονται σε ορισμένα συστήματα
  - Αρχείο που φαίνεται να έχει φορτωθεί στη μνήμη
    - Χρήση με λειτουργίες μνήμης αντί εισόδου/εξόδου
    - Το αρχείο μπορεί να σελιδοποιείται όπως όλη η μνήμη
    - Έτσι μπορούν να φορτώνονται και τα DLL
  - Επιτρέπει και επικοινωνία διεργασιών
    - Πολλές διεργασίες χαρτογραφούν το ίδιο αρχείο

Πολιτική καθαρισμού (1 από 2)

- Πολιτική καθαρισμού
  - Οι τροποποιημένες σελίδες δεν είναι διαθέσιμες
    - Πρέπει να περιμένουμε να γραφτούν στο δίσκο
    - Καθυστέρηση όταν θέλουμε να φορτώσουμε νέες σελίδες
  - Δαίμονας σελιδοποίησης
    - Φροντίζει να υπάρχουν πάντα διαθέσιμες σελίδες
  - Ο δαίμονας ξυπνάει περιοδικά
    - Αν έχουμε λίγες διαθέσιμες επιλέγει μερικές για αφαίρεση
    - Αν είναι τροποποιημένες τις στέλνει και για εγγραφή

Πολιτική καθαρισμού (2 από 2)

- Πολιτική καθαρισμού
  - Ο δαίμονας δεν διώχνει τις σελίδες που επιλέγει
    - Αν ζητηθούν ξανά, είναι ακόμα διαθέσιμες
    - Τώρα όμως θα είναι και καθαρές
    - Άρα μπορούν να χρησιμοποιηθούν άμεσα
- Πολιτική ρολογιού με δύο δείκτες
  - Ο εμπρός δείκτης επιλέγει σελίδες για εγγραφή
  - Φροντίζει να υπάρχουν διαθέσιμες σελίδες αν χρειαστούν
  - Ο πίσω δείκτης επιλέγει σελίδες προς αντικατάσταση
  - Χρησιμοποιεί τις διαθέσιμες σελίδες για φόρτωση νέων

Διασύνδεση εικονικής μνήμης

- Η κλασική διασύνδεση είναι αυτή που περιγράψαμε
  - Μπορεί όμως να επιτρέπονται και αποκλίσεις
- Παράδειγμα: κοινή μνήμη
  - Απόδοση ονόματος σε ορισμένες περιοχές
  - Χρήση ονόματος για σύνδεση με διαφορετικές διεργασίες
  - Η κοινή μνήμη επιτρέπει γρήγορη επικοινωνία διεργασιών
- Παράδειγμα: κατανεμημένη κοινόχρηστη μνήμη
  - Κοινός χώρος διευθύνσεων σε χωριστές μηχανές
  - Σφάλμα σελίδας όταν μια μηχανή δεν έχει τη σελίδα
  - Οι σελίδες στέλνονται από μηχάνημα σε μηχάνημα

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Ζητήματα υλοποίησης

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 3:** Διαχείριση Μνήμης

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ρόλος του ΛΣ (1 από 2)

- Λειτουργικό σύστημα και σελιδοποίηση
  - Δημιουργία νέας διεργασίας
    - Εκτίμηση μεγέθους προγράμματος και δεδομένων
    - Δημιουργία και αρχικοποίηση πίνακα σελίδων
    - Δέσμευση χώρου στην περιοχή εναλλαγής
    - Ενημέρωση πίνακα διεργασιών
  - Χρονοπρογραμματισμός διεργασίας
    - Καθαρισμός MMU και TLB
    - Αλλαγή πίνακα σελίδων
    - Προαιρετικά φόρτωση ορισμένων σελίδων

Ρόλος του ΛΣ (2 από 2)

- Λειτουργικό σύστημα και σελιδοποίηση
  - Σφάλμα σελίδας
    - Εντοπισμός ζητούμενης σελίδας στο δίσκο
    - Φόρτωση σελίδας σε διαθέσιμο πλαίσιο
    - Επαναφορά μετρητή προγράμματος και επανεκτέλεση
  - Τερματισμός διεργασίας
    - Αποδέσμευση πίνακα σελίδων
    - Αποδέσμευση χώρου στην περιοχή εναλλαγής
    - Αποδέσμευση μη καταμεριζόμενων σελίδων

Χειρισμός σφαλμάτων σελίδας (1 από 3)

- Χειρισμός σφαλμάτων σελίδας
  - Το υλικό προκαλεί παγίδα στον πυρήνα
    - Ο μετρητής προγράμματος αποθηκεύεται στη στοίβα
  - Εκτέλεση βασικού χειριστή διακοπής
    - Αποθήκευση υπόλοιπων καταχωρητών στη στοίβα
    - Κλήση του χειριστή σφαλμάτων σελίδας
  - Χειριστής σφαλμάτων σελίδας
    - Καθορισμός ζητούμενης σελίδας
    - Μπορεί να απαιτεί ανάλυση της εντολής που διακόπηκε

Χειρισμός σφαλμάτων σελίδας (2 από 3)

- Χειρισμός σφαλμάτων σελίδας
  - Έλεγχος εγκυρότητας και πρόσβασης
    - Αν υπάρχει πρόβλημα στέλνεται μήνυμα στη διεργασία
    - Επιλέγεται ένα ελεύθερο πλαίσιο ή εκτελείται η αντικατάσταση
  - Καθαρισμός πλαισίου σελίδας (αν είναι «βρώμικο»)
    - Χρονοπρογραμματίζεται η εγγραφή της σελίδας στο δίσκο
    - Η διεργασία αναστέλλεται
  - Φόρτωση της σελίδας στη μνήμη
    - Χρονοπρογραμματίζεται η ανάγνωση της σελίδας από το δίσκο
    - Η διεργασία αναστέλλεται (αν δεν είναι ήδη σε αναστολή)

Χειρισμός σφαλμάτων σελίδας (3 από 3)

- Χειρισμός σφαλμάτων σελίδας
  - Ενημέρωση πίνακα σελίδων
    - Μόλις ολοκληρωθεί η μεταφορά της σελίδας
  - Επαναφορά μετρητή στην εντολή που διακόπηκε
  - Χρονοπρογραμματισμός της διεργασίας
    - Θα ξεκινήσει με την εντολή που διακόπηκε
  - Επαναφορά των καταχωρητών
    - Στην κατάσταση που ήταν πριν τη διακοπή

Αντίγραφα ασφαλείας εντολών (1 από 2)

- Αντίγραφα ασφαλείας εντολών
  - Η επανεκτέλεση της εντολής δεν είναι πάντα απλή
  - Παράδειγμα: MOVE.L #6(A1),2(A0) [Motorola 68K]
  - Εντολή 6 byte με 2 σχετικές διευθύνσεις
  - Η αιτία του σφάλματος σελίδας δεν είναι προφανής
    - Το σφάλμα μπορεί να συμβεί κατά τις τρεις προσκομίσεις
    - Επιπλέον μπορεί να συμβεί στις δύο αναφορές στη μνήμη

Αντίγραφα ασφαλείας εντολών (2 από 2)

- Χρήση ειδικών καταχωρητών στον επεξεργαστή
  - Καταχωρητής αρχικής διεύθυνσης τρέχουσας εντολής
    - Χρήσιμος αν έχουμε εντολές με διάφορα μεγέθη
  - Καταχωρητής αλλαγών σε καταχωρητές
  - Βοηθούν στον εντοπισμό της αιτίας του σφάλματος
  - Χρησιμοποιούνται για αναίρεση όλων των αλλαγών
  - Η εντολή μπορεί να ξεκινήσει από την αρχή
    - Με τις ίδιες συνθήκες όπως στην αρχική εκτέλεση

Κλείδωμα σελίδων στη μνήμη

- Κλείδωμα σελίδων στη μνήμη
  - Έστω ότι μια διεργασία ξεκινά ανάγνωση του δίσκου
    - Η διεργασία μπλοκάρεται περιμένοντας τα δεδομένα
  - Τι θα γίνει αν η σελίδα δεδομένων αντικατασταθεί;
    - Αν χρησιμοποιούμε DMA θα γράψουμε στο λάθος σημείο!
  - Οι σελίδες για είσοδο-έξοδο κλειδώνονται
    - Καρφίτσωμα (pinning) σελίδων στη μνήμη
  - Εναλλακτικά είσοδος-έξοδος μόνο μέσω πυρήνα

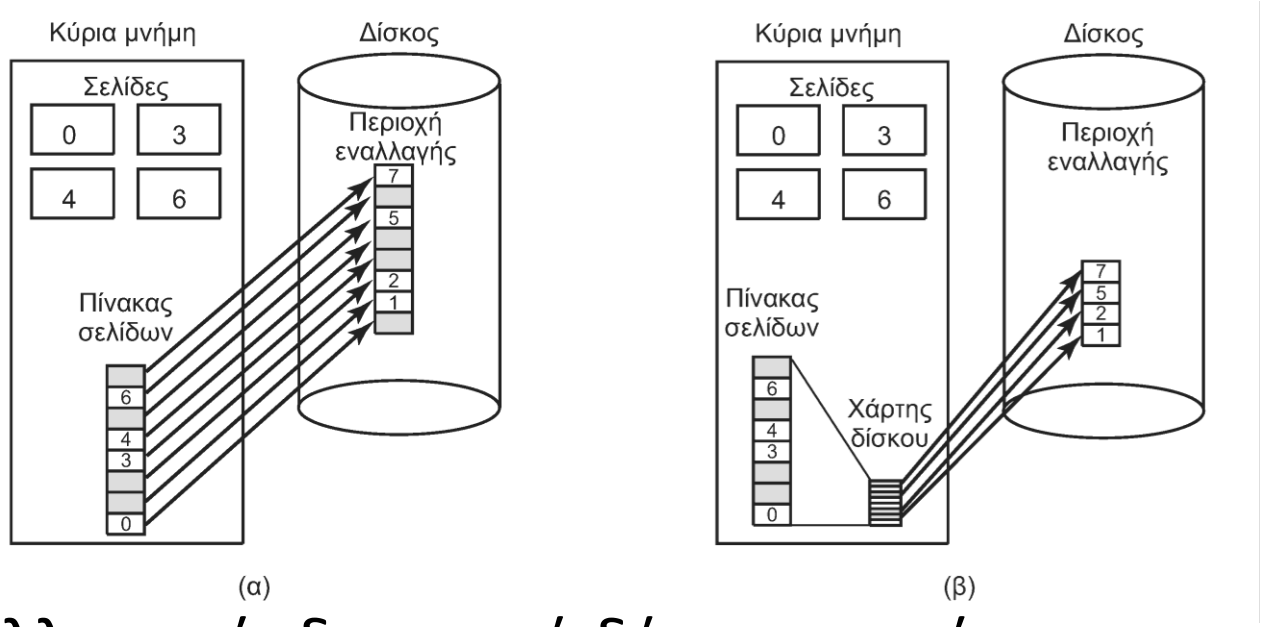
Δευτερεύουσα μνήμη (1 από 4)

- Διαμέριση ή δίσκος εναλλαγής (swap)
  - Ο δίσκος έχει μεγαλύτερη ταχύτητα προσπέλασης
  - Η περιοχή εναλλαγής δεν περιέχει σύστημα αρχείων
    - Το ΛΣ αναφέρεται απευθείας στα μπλοκ στο δίσκο
  - Κατά την εκκίνηση η περιοχή είναι κενή
  - Σε κάθε διεργασία εκχωρείται μέρος της περιοχής
  - Συνεχόμενη περιοχή ανάλογα με μέγεθος διεργασίας
    - Απαιτεί διαχείριση ελεύθερων περιοχών

Δευτερεύουσα μνήμη (2 από 4)

- Διαμέριση ή δίσκος εναλλαγής (swap)
  - Για κάθε διεργασία θυμόμαστε πού ξεκινά η περιοχή της
    - Αποθήκευση στην καταχώριση του πίνακα διεργασιών
    - Εύκολη αντιστοίχιση σελίδων σε μπλοκ του δίσκου
  - Αρχικοποίηση της περιοχής ανταλλαγής
    - Είτε αντιγραφή της εικόνας στην περιοχή πρώτα
    - Είτε αντιγραφή της εικόνας στη μνήμη πρώτα
  - Τι γίνεται αν η διεργασία μεγαλώσει;
    - Δέσμευση ξεχωριστών περιοχών για κώδικα και δεδομένα
    - Δέσμευση πολλών τμημάτων για τα δεδομένα

Δευτερεύουσα μνήμη (3 από 4)



- Εναλλακτικά, δυναμική δέσμευση χώρου
  - Χρησιμοποιούμε χώρο μόνο όταν μια σελίδα εκτοπίζεται
  - Απαιτεί πίνακα με διευθύνσεις σελίδων στη μνήμη
  - Πιο οικονομική και ευέλικτη από τη σταθερή περιοχή
  - Απαιτεί διαχείριση του πίνακα διευθύνσεων στο δίσκο

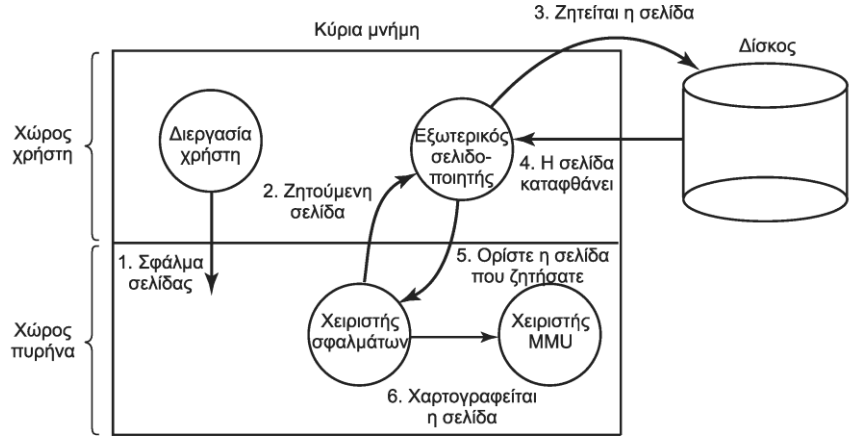
Δευτερεύουσα μνήμη (4 από 4)

- Χρήση αρχείων ως περιοχών ανταλλαγής
  - Χρησιμοποιούνται στα Windows
    - Επιτρέπονται και διαμερίσεις
  - Πιο ευέλικτη διαχείριση χώρου
  - Πιο αργή αντιστοίχιση
    - Δεν παρακάμπτεται το σύστημα αρχείων
  - Χρήση αρχείου προγράμματος για τον κώδικα
    - Βρίσκεται ήδη αποθηκευμένο στο δίσκο
    - Αυτό επιτρέπεται και με διαμερίσεις

Πολιτική και μηχανισμός (1 από 3)

- Διαχωρισμός πολιτικής και μηχανισμού
  - Εκτέλεση διαχειριστή μνήμης σε επίπεδο χρήστη
    - Ο πυρήνας παρέχει το μηχανισμό
    - Ο διαχειριστής παρέχει την πολιτική
    - Χρησιμοποιήθηκε στο σύστημα Mach
  - Χειριστής χαμηλού επιπέδου της MMU (πυρήνας)
    - Γράφεται σε assembly ανάλογα με τη μηχανή
  - Χειριστής σφαλμάτων σελίδας (πυρήνας)
    - Γράφεται σε C ανάλογα με το λειτουργικό σύστημα

Πολιτική και μηχανισμός (2 από 3)



- Εξωτερικός σελιδοποιητής (επίπεδο χρήστη)
  - Δημιουργεί τον πίνακα σελίδων και δεσμεύει χώρο
  - Ενημερώνεται από τον χειριστή σφαλμάτων σελίδας
  - Διαβάζει τις σελίδες από το δίσκο στη μνήμη του
  - Ενημερώνει χειριστή MMU όταν ολοκληρωθεί η φόρτωση
  - Ο χειριστής MMU απεικονίζει τις σελίδες στην διεργασία

Πολιτική και μηχανισμός (3 από 3)

- Πού βρίσκεται ο αλγόριθμος αντικατάστασης;
  - Στον εξωτερικό σελιδοποιητή
    - Πρέπει να του μεταφέρονται τα bit A και T με κάποιο τρόπο
  - Στον πυρήνα
    - Επιλέγει τη σελίδα προς αντικατάσταση
    - Απεικονίζει τη σελίδα στο χώρο του σελιδοποιητή
    - Ο σελιδοποιητής γράφει τη σελίδα στο δίσκο (αν χρειάζεται)
    - Ο σελιδοποιητής διαβάζει τη ζητούμενη σελίδα
- Ο εξωτερικός σελιδοποιητής παρέχει μεγάλη ευελιξία
  - Μπορεί να εκτελεί αλγόριθμους ανάλογα με τη διεργασία

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

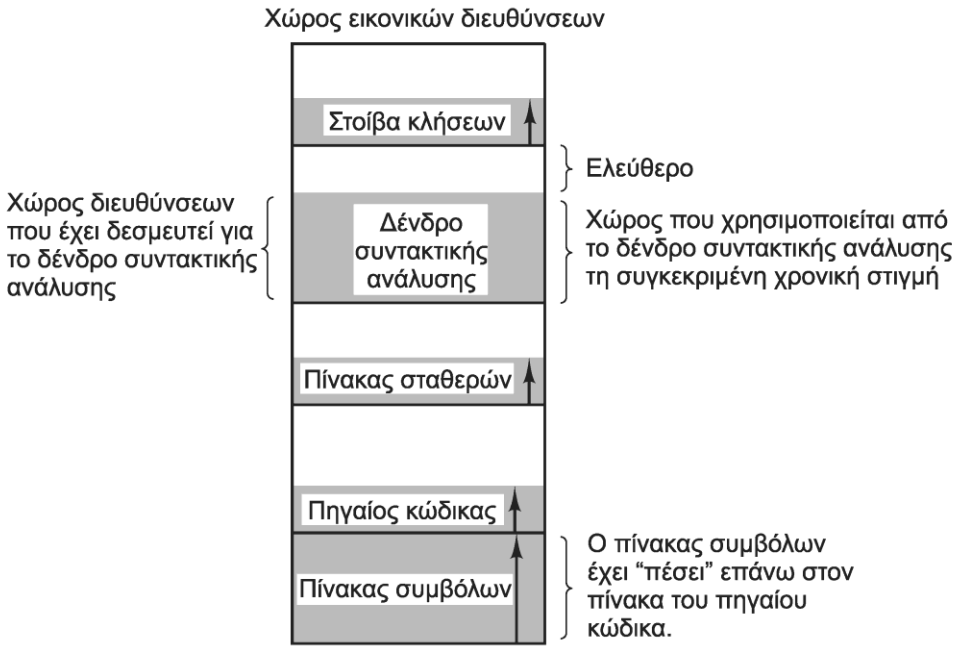
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Τμηματοποίηση

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 3:** Διαχείριση Μνήμης

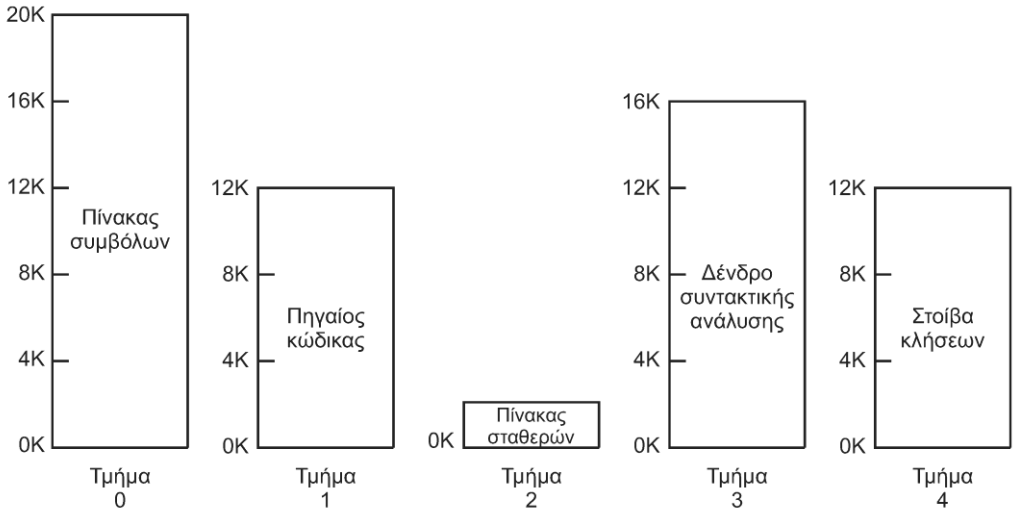
Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Τμηματοποίηση (1 από 4)



- Πολλοί εικονικοί χώροι διευθύνσεων
  - Παράδειγμα: μεταγλωττιστής
    - Περιέχει πολλούς δυναμικούς πίνακες
    - Κάποιοι πίνακες μπορεί να γεμίσουν ενώ άλλοι όχι

Τμηματοποίηση (2 από 4)



- Τμηματοποίηση (segmentation)
  - Κάθε διεργασία αποτελείται από τμήματα (segments)
  - Κάθε τμήμα είναι μια γραμμική ακολουθία διευθύνσεων
  - Τα τμήματα έχουν διαφορετικό και μεταβλητό μέγεθος
  - Λογικός καταμερισμός της μνήμης σε τμήματα

Τμηματοποίηση (3 από 4)

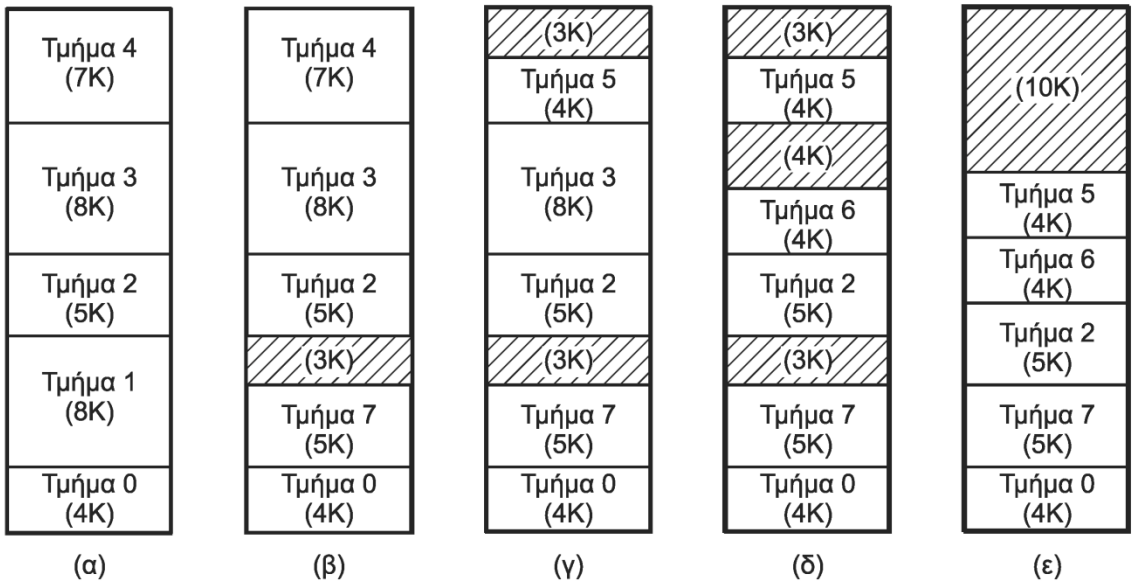
- Πλεονεκτήματα τμηματοποίησης
  - Απλούστερη σύνδεση διαδικασιών
    - Κάθε διαδικασία είναι διαφορετικό τμήμα
    - Δεν χρειάζεται επανατοποθέτηση κώδικα κατά τη σύνδεση
  - Διευκόλυνση κοινής χρήσης βιβλιοθηκών
    - Κάθε βιβλιοθήκη είναι διαφορετικό τμήμα
    - Όλες οι διεργασίες τη βλέπουν με τον ίδιο τρόπο
  - Προστασία μνήμης σε επίπεδο τμήματος
    - Τα τμήματα είναι λογικές οντότητες (όχι όπως οι σελίδες)
    - Μπορούν να προστατεύονται ανάλογα με τη φύση τους

Τμηματοποίηση (4 από 4)

Χρειάζεται να είναι ενήμερος ο προγραμματιστής ότι χρησιμοποιείται η συγκεκριμένη τεχνική;	Όχι	Ναι
Πόσοι γραμμικοί χώροι διευθύνσεων υπάρχουν;	1	Πολλοί
Μπορεί ο συνολικός χώρος διευθύνσεων να υπερβαίνει το μέγεθος της φυσικής μνήμης;	Ναι	Ναι
Μπορούν οι διαδικασίες και τα δεδομένα να είναι διακριτά και να προστατεύονται ξεχωριστά;	Όχι	Ναι
Είναι εύκολος ο χειρισμός πινάκων μεταβλητού μεγέθους;	Όχι	Ναι
Διευκολύνεται η κοινή χρήση των διαδικασιών ανάμεσα στους χρήστες;	Όχι	Ναι
Γιατί επινοήθηκε αυτή η τεχνική;	Για να δημιουργηθεί μεγάλος και γραμμικός χώρος διευθύνσεων χωρίς να χρειάζεται να επεκταθεί η φυσική μνήμη	Για να επιτραπεί στα προγράμματα και τα δεδομένα να χωρίζονται σε λογικά ανεξάρτητους χώρους διευθύνσεων, και να υποβοηθείται η κοινοχρησία και η προστασία

Σύγκριση σελιδοποίησης και τμηματοποίησης

Υλοποίηση τμηματοποίησης

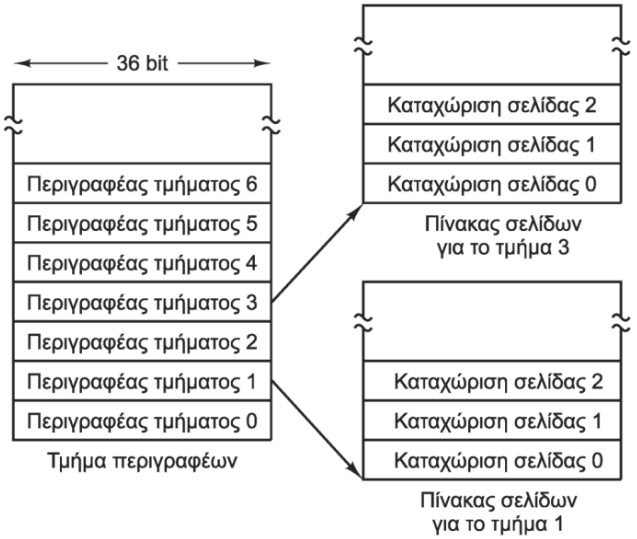


- Υλοποίηση αμιγούς τμηματοποίησης
  - Τα τμήματα έχουν μεταβλητό μέγεθος
  - Η φόρτωση / αφαίρεση τμημάτων αφήνει κενά στη μνήμη
    - Εξωτερική κατάτμηση: κενά ανάμεσα στα τμήματα μνήμης
  - Ίδια προβλήματα όπως με τους καταχωρητές βάσης

Υλοποίηση: MULTICS (1 από 6)

- Τμηματοποίηση με σελιδοποίηση: MULTICS
  - Κάθε τμήμα μπορεί να σελιδοποιείται
    - Φόρτωση μέρους μεγάλων τμημάτων στη μνήμη
    - Απλούστευση της διαχείρισης μνήμης με τις σελίδες
  - Το MULTICS σχεδιάστηκε στα Honeywell 6000
    - 2^{18} τμήματα με 2^{16} λέξεις των 36 bit ανά διεργασία
  - Κάθε διεργασία διαθέτει πίνακα τμημάτων
    - Ένας περιγραφέας (descriptor) ανά τμήμα
    - Ο πίνακας είναι κι αυτός τμήμα που σελιδοποιείται

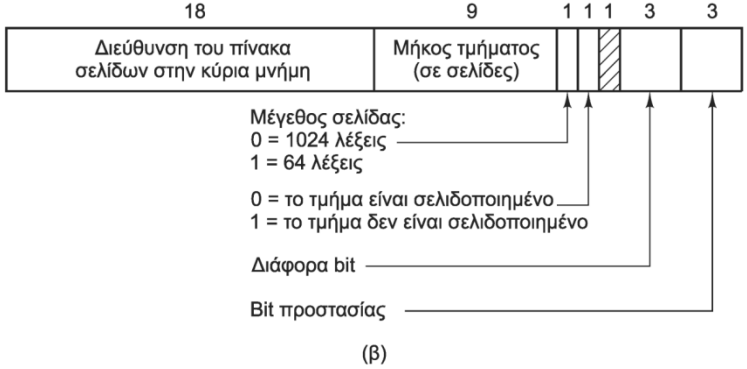
Υλοποίηση: MULTICS (2 από 6)



(α)

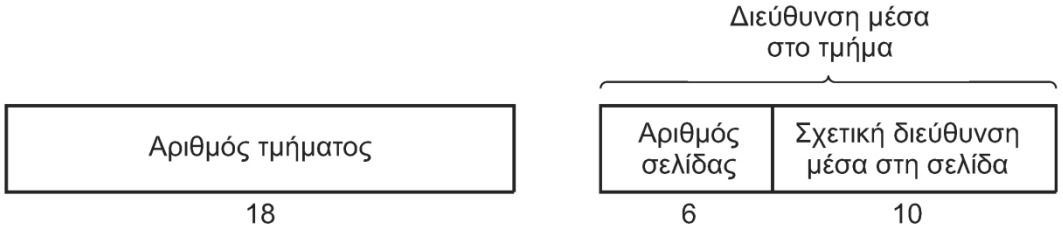
- Τμηματοποίηση με σελιδοποίηση: MULTICS
  - Ο περιγραφέας δείχνει στον πίνακα σελίδων τμήματος
    - Ο δείκτης έχει μέγεθος 18 bit ενώ οι διευθύνσεις 24 bit
    - Στο τέλος του δείκτη προστίθεται το 000000
    - Ουσιαστικά οι σελίδες ευθυγραμμίζονται ανά 64 byte

Υλοποίηση: MULTICS (3 από 6)



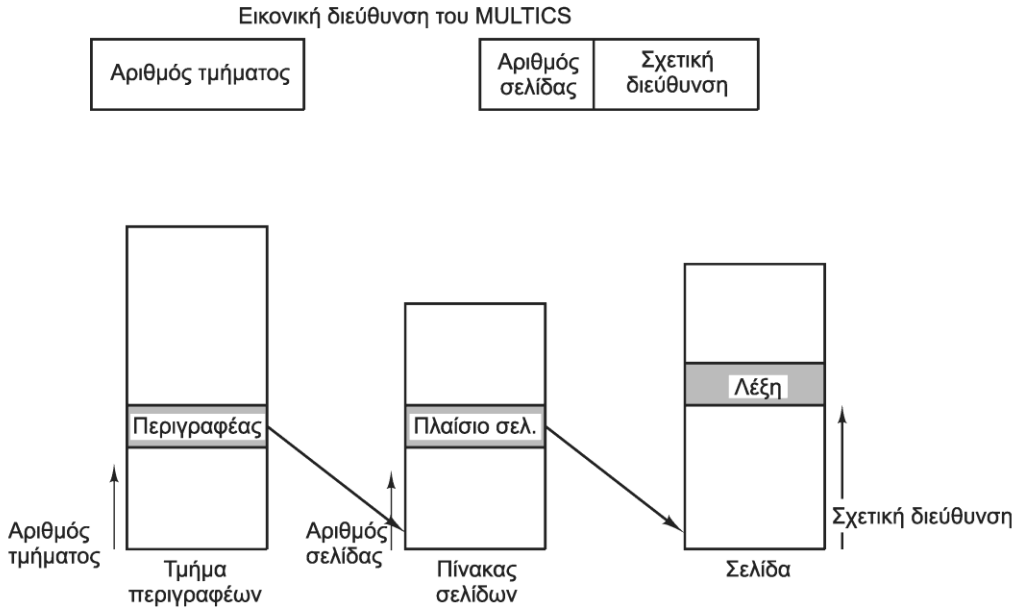
- Τμηματοποίηση με σελιδοποίηση: MULTICS
  - Μέγεθος σελίδας: 1024 ή 64 λέξεις (μικρά τμήματα)
  - Σελιδοποίηση ή όχι (ορισμένα τμήματα του MULTICS)
  - Bit προστασίας τμήματος
  - Η διεύθυνση στη δευτερεύουσα μνήμη δεν περιέχεται
    - Βρίσκεται σε πίνακα του χειριστή σφαλμάτων τμήματος

Υλοποίηση: MULTICS (4 από 6)



- Τμηματοποίηση με σελιδοποίηση: MULTICS
  - Οι διευθύνσεις αποτελούνται από δύο μέρη
  - Η απόσταση χωρίζεται και αυτή σε δύο μέρη
  - Μετάφραση εικονικών σε φυσικές διευθύνσεις
    - Εντοπισμός περιγραφέα τμήματος από τον αριθμό του
    - Έλεγχος πρόσβασης και παρουσίας τμήματος στη μνήμη
    - Έλεγχος αν η σελίδα βρίσκεται στη μνήμη
    - Πρόσθεση απόστασης στην αρχή της σελίδας

Υλοποίηση: MULTICS (5 από 6)



- Τμηματοποίηση με σελιδοποίηση: MULTICS
  - Και ο πίνακας τμημάτων είναι σελιδοποιημένος
    - Ένας καταχωρητής δείχνει στην αρχή του
  - Στη διεργασία μπορούν να συμβούν διάφορα σφάλματα
    - Προστασίας, τμήματος ή σελίδας

Υλοποίηση: MULTICS (6 από 6)

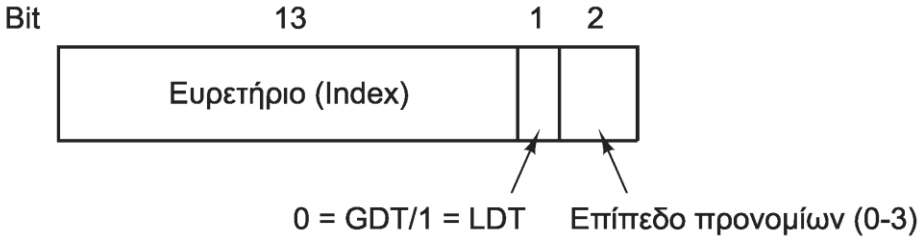
Πεδίο σύγκρισης					Χρησιμοποιείται αυτή η καταχώριση;
Αριθμός τμήματος	Εικονική σελίδα	Πλαίσιο σελίδας	Προστασία	Ηλικία	
4	1	7	Ανάγνωση/Εγγραφή	13	1
6	0	2	Μόνο ανάγνωση	10	1
12	3	1	Ανάγνωση/Εγγραφή	2	1
					0
2	1	0	Μόνο εκτέλεση	7	1
2	2	12	Μόνο εκτέλεση	9	1

- Τμηματοποίηση με σελιδοποίηση: MULTICS
  - Χρήση TLB 16 λέξεων για επιτάχυνση των μεταφράσεων
  - Αν βρεθεί το τμήμα/σελίδα πάμε απευθείας στη μνήμη
  - Αν δεν βρεθεί η καταχώριση, καταφεύγουμε στους πίνακες
    - Η καταχώριση φορτώνεται στο TLB στη θέση της παλιότερης
    - Πεδίο ηλικίας που δείχνει πόσο παλιά είναι κάθε καταχώριση

Υλοποίηση: Pentium (1 από 8)

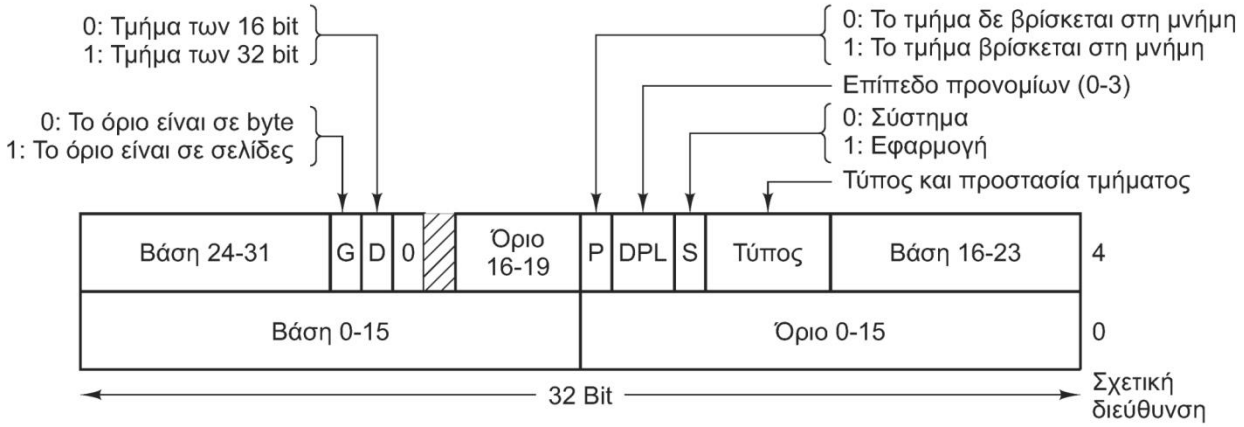
- Τμηματοποίηση με σελιδοποίηση: Intel Pentium
  - Λιγότερα τμήματα (2^{14}), μεγαλύτερο μέγεθος (2^{20} ή 2^{32})
  - Κάθε διεργασία βλέπει δύο πίνακες τμημάτων
    - Πίνακας τοπικών περιγραφέων (LDT) ανά διεργασία
    - Περιέχει τον κώδικα και τα δεδομένα της διεργασίας
    - Πίνακας καθολικών περιγραφέων (GDT) για όλες τις διεργασίες
    - Περιέχει το λειτουργικό και τη μνήμη του συστήματος
  - Ο Pentium διαθέτει έξι καταχωρητές τμημάτων
    - CS: τμήμα κώδικα, DS: τμήμα δεδομένων, SS: τμήμα στοίβας
    - Φορτώνονται με επιλογείς (selectors) τμημάτων των 16 bit

Υλοποίηση: Pentium (2 από 8)



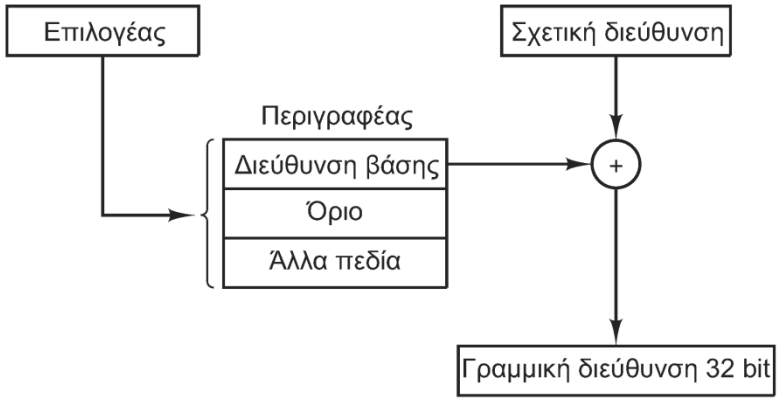
- Τμηματοποίηση με σελιδοποίηση: Intel Pentium
  - Περιεχόμενα επιλογέα τμήματος
    - Τοπικό ή καθολικό τμήμα (δηλαδή, LDT ή GDT)
    - Θέση στον πίνακα (13 bit, άρα 2^{13} τμήματα ανά πίνακα)
  - Μαζί με τον επιλογέα φορτώνεται και ο περιγραφέας
    - Αποθηκεύεται σε έναν εσωτερικό καταχωρητή
    - Οι περιγραφείς έχουν μήκος 8 byte ο καθένας
    - Αντιγράφεται από τον LDT ή τον GDT
    - Χρήση του επιλογέα με μηδενισμένα τα 3 τελευταία bit

Υλοποίηση: Pentium (3 από 8)



- Τμηματοποίηση με σελιδοποίηση: Intel Pentium
  - Ο περιγραφέας είναι περίεργος για λόγους συμβατότητας
    - Τα τμήματα μπορεί να αρχίζουν οπουδήποτε στη μνήμη
  - Το όριο ερμηνεύεται διαφορετικά ανάλογα με το πεδίο G
  - Το πεδίο P δείχνει αν το τμήμα είναι στο μνήμη
  - Το πεδίο S δείχνει αν το τμήμα ανήκει στο σύστημα
  - Το πεδίο DPL περιέχει το πεδίο προστασίας τμήματος

Υλοποίηση: Pentium (4 από 8)

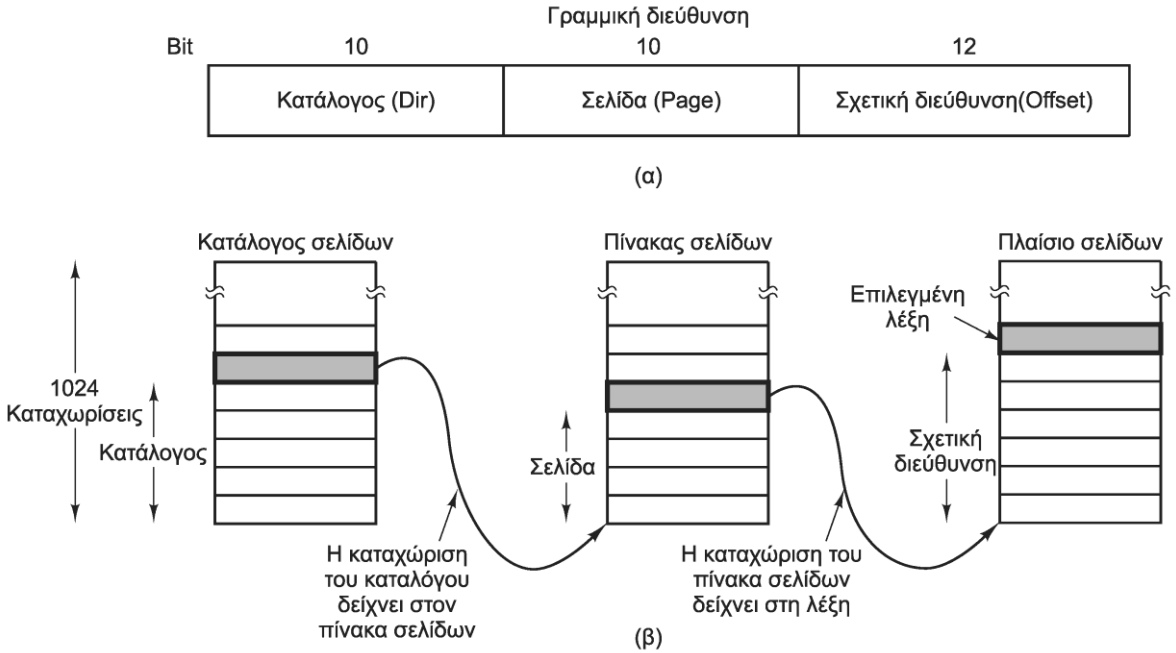


- Μετάφραση λογικής διεύθυνσης σε φυσική
  - Επιλογή περιγραφέα με βάση τον καταχωρητή τμήματος
  - Σφάλμα τμήματος αν το τμήμα δεν είναι στη μνήμη
  - Σύγκριση απόστασης με όριο τμήματος
  - Πρόσθεση απόστασης στη βάση του τμήματος
  - Προκύπτει μια γραμμική (linear) διεύθυνση των 32 bit
    - Μπορεί να είναι σελιδοποιημένη ή όχι

Υλοποίηση: Pentium (5 από 8)

- Τμηματοποίηση με σελιδοποίηση: Intel Pentium
  - Πόσο μεγάλα είναι τα τμήματα;
    - Εξαρτάται από το πεδίο G του περιγραφέα
    - Είτε 2^{20} byte (όπως στον 8086/8088) είτε 2^{20} σελίδες
  - Οι πίνακες σελίδων είναι δύο επιπέδων
  - Κάθε διεργασία έχει έναν κατάλογο σελίδων
    - Κάθε θέση του δείχνει σε πίνακα δεύτερου επιπέδου
    - Κάθε πίνακας είναι μία σελίδα με 2^{10} καταχωρήσεις των 32 bit
  - Τα 20 bit κάθε καταχώρισης δείχνουν ένα πλαίσιο σελίδας
    - Τα υπόλοιπα bit δείχνουν την κατάσταση της σελίδας

Υλοποίηση: Pentium (6 από 8)

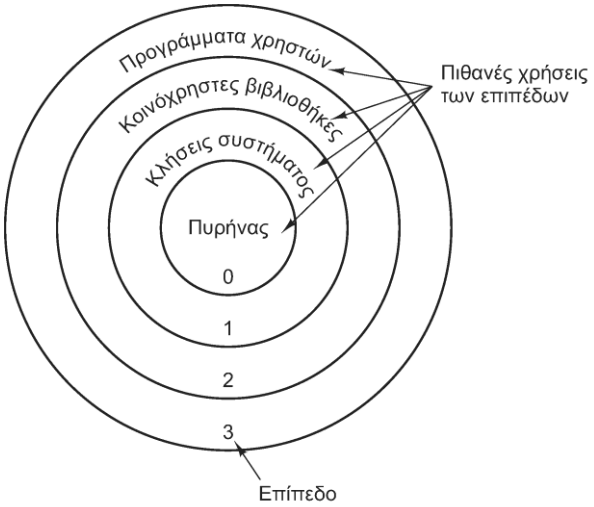


- Οι γραμμικές διευθύνσεις χωρίζονται σε τρία μέρη
  - Δείκτης στον κατάλογο σελίδων(10 bit)
  - Δείκτης στον πίνακα σελίδων δεύτερου επιπέδου (10 bit)
  - Απόσταση μέσα στη σελίδα (12 bit)

Υλοποίηση: Pentium (7 από 8)

- Τμηματοποίηση με σελιδοποίηση: Intel Pentium
  - Χρήση TLB για παράκαμψη των πινάκων σελίδων
  - Η τμηματοποίηση μπορεί να απενεργοποιηθεί
    - Όλοι οι καταχωρητές τμημάτων ορίζονται ίδιοι
- Μηχανισμός προστασίας του Pentium
  - Τέσσερα επίπεδα προνομίων, το 0 είναι πιο προνομιούχο
  - Κάθε τμήμα μνήμης ανήκει σε ένα επίπεδο
    - Σύμφωνα με επιλογέα και περιγραφέα
  - Η εκτελούμενη διεργασία ανήκει και αυτή σε ένα επίπεδο
    - Ειδικό πεδίο στον καταχωρητή κατάστασης

Υλοποίηση: Pentium (8 από 8)



- Μηχανισμός προστασίας του Pentium
  - Ελεγχόμενη πρόσβαση σε κώδικα χαμηλότερου επιπέδου
    - Οι εντολές κλήσης αναφέρονται σε επιλογέα/περιγραφέα
    - Επιτρέπεται να μεταβούμε μόνο σε πύλες κλήσης (call gates)
    - Οι πύλες κλήσης επιλέγονται από το κατώτερο επίπεδο
  - Ίδιος μηχανισμός και για παγίδες και διακοπές

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Τέλος Ενότητας #3

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 3:** Διαχείριση Μνήμης

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής