

# Λογισμικό διαχείρισης μνήμης (Memory management software)

## ΣΤΟΧΟΙ

- Υλοποίηση εικονικής μνήμης
  - Λογική διαίρεση του εικονικού χώρου διευθύνσεων μιας διεργασίας
  - Δυναμική μεταθεσιμότητα
  - Προστασία
- Αποδοτική χρήση της φυσικής μνήμης

ΤΕΜΑΧΙΣΜΟΣ	ΣΕΛΙΔΟΠΟΙΗΣΗ
Λογική διαίρεση του εικονικού χώρου διευθύνσεων του προγράμματος	Φυσική διαίρεση του χώρου διευθύνσεων της κύριας μνήμης
Από το μεταγλωττιστή /διερμηνευτή/προγραμματιστή (μ/δ/π)	Από το υλισμικό
<b>Τμήμα</b> - Λέξη ( $s - w$ )	<b>Σελίδα</b> - Λέξη ( $p - w$ )
Προστασία: Όριο/Μήκος	
Μεταβλητό μέγεθος τμήματος (μ/δ/π)	Σταθερό μέγεθος σελίδας (αρχιτεκτονική μηχανής)
Όνομα τμήματος (μ/δ/π) ΛΣ: Όνομα τμήματος → Αριθμός τμήματος	Αριθμός σελίδας (ΛΣ)

# Λογισμικό διαχείρισης μνήμης

## ΤΕΜΑΧΙΟ

- σταθερού μεγέθους = σελίδα
- μεταβλητού μεγέθους = τμήμα

## ΚΑΘΗΚΟΝΤΑ ΔΙΑΧΕΙΡΙΣΤΗ ΜΝΗΜΗΣ

- **Διατηρεί δομές δεδομένων**
  - Παρουσία τεμαχίων κάθε διεργασίας στην κύρια μνήμη (ΚΜ) ή όχι
  - Θέση τεμαχίων στην κύρια, στη βοηθητική μνήμη (ΒΜ) και/ή στην περιοχή ανταλλαγής
  - Θέσεις ελεύθερων και κατειλημμένων περιοχών της ΚΜ
- **Εφαρμόζει τις πολιτικές:**
  - **Προσκόμισης (Fetch policy)**
    - Μεταφορά τεμαχίων από τη ΒΜ στην ΚΜ
    - ΠΟΙΑ τεμάχια θα μεταφερθούν?
    - ΠΟΤΕ θα μεταφερθούν?
  - **Τοποθέτησης (Placement)**
    - Τοποθέτηση στην ΚΜ των προσκομιζόμενων τεμαχίων
    - ΠΟΙΕΣ ΘΕΣΕΙΣ θα καταλάβουν?
  - **Αντικατάστασης (Replacement)**
    - Μεταφορά τεμαχίων από την ΚΜ στη ΒΜ (αν τα προσκομιζόμενα τεμάχια δεν χωράνε)
    - ΠΟΙΑ τεμάχια θα μεταφερθούν?
- Ενημερώνει τις δομές δεδομένων του για κάθε μεταβολή

# Λογισμικό διαχείρισης μνήμης

- **Πολιτικές προσκόμισης και αντικατάστασης**
  - Το ίδιο σύνθετες για τεμαχισμό και σελιδοποίηση
- **Πολιτική τοποθέτησης**
  - **σελιδοποίηση: ασήμαντη**
    - μνήμη → ισομεγέθη πλαίσια
    - κατάλληλη ενημέρωση του πινάκων σελίδων
    - ελεύθερες περιοχές της μνήμης = κενά πλαίσια μνήμης
  - **τεμαχισμός: σύνθετη**
    - ελεύθερες περιοχές μνήμης μεταβλητού μεγέθους
    - δε χωράνε όλες το τεμάχιο που πρόκειται να προσκομισθεί

**Κατάσταση (Status/State) τεμαχίου** = σύνολο δυφίων

- **θέση τεμαχίου**
  - στην KM, στη BM, στην περιοχή ανταλλαγής ή μεταφέρεται
- **λερωμένο(dirty) ή αναλλοίωτο/καθαρό (clean)**
  - ένα καθαρό τεμάχιο δε χρειάζεται να μεταφερθεί στη BM
- **έχει γίνει αναφορά (reference) στο τεμάχιο** ή όχι  
(από τότε που φορτώθηκε στη μνήμη)
- **(κατα)μεριζόμενο (shared) ή όχι**

# Λογισμικό διαχείρισης μνήμης

## Περιγραφητής τεμαχίου

- αριθμός τεμαχίου
- δυφία κατάστασης
- διεύθυνση στην περιοχή ανταλλαγής
- διεύθυνσή στη BM
- προνόμια προσπέλασης
- διεύθυνση βάσης/αριθμός πλαισίου στην KM
- όριο ή μήκος [για τμήμα]
- ταυτότητα της διεργασίας που χρησιμοποιεί το τεμάχιο
- πλήθος διεργασιών που το χρησιμοποιούν

## Πίνακας Τεμαχίων (Section Table, ST)

- ξεχωριστός ανά διεργασία ή κεντρικός για όλες τις διεργασίες
- κεντρικός ST μπορεί ν' αποτελεί ένα ολόκληρο τμήμα = τμήμα περιγραφητή
- που πρέπει να είναι μόνιμα φορτωμένο στη μνήμη και ο
- καταχωρητής βάσης τμήματος περιγραφητή
  - φορτώνεται με τη διεύθυνση βάσης του τμήματος περιγραφητή

# Λογισμικό διαχείρισης μνήμης

## ΚΑΤΕΙΛΗΜΜΕΝΕΣ-ΕΛΕΥΘΕΡΕΣ ΠΕΡΙΟΧΕΣ

### Σελιδοποίηση

- χάρτης μνήμης (memory/core map) ή χάρτης δυφίων (bit map)
  - 1 δυφίο για κάθε πλαίσιο μνήμης
  - ελεύθερο/κατειλημμένο πλαίσιο
  - μέγεθος αντιστρόφως ανάλογο του μεγέθους των πλαισίων

### Τεμαχισμός

- χάρτης δυφίων?
- δύο διαφορετικοί πίνακες/λίστες μεταβλητού μεγέθους
  - λίστα ελευθέρων περιοχών (free list) περιοχών
    - διεύθυνση βάσης της περιοχής
    - μήκος της περιοχής
    - ταξινομημένη
      - είτε με τη διεύθυνση βάσης της περιοχής
      - είτε με το μέγεθος της περιοχής

# Λογισμικό διαχείρισης μνήμης

## ΚΑΤΕΙΛΗΜΜΕΝΕΣ-ΕΛΕΥΘΕΡΕΣ ΠΕΡΙΟΧΕΣ

- **λίστα κατειλημμένων (occupied list) περιοχών**
  - διεύθυνση βάσης της περιοχής
  - μήκος της περιοχής
  - ταυτότητα της διεργασίας που χρησιμοποιεί την περιοχή
- **μια διπλά συνδεδεμένη λίστα**
  - διεύθυνση βάσης περιοχής
  - όριο (ή μήκος) περιοχής
  - δυφίο ελεύθερης/κατειλημμένης περιοχής
  - δείκτης/ες από τον περιγραφητή κάθε διεργασίας
    - προς κόμβο/ους (κατειλημμένων περιοχών) που αντιστοιχούν στη διεργασία

# Λογισμικό διαχείρισης μνήμης

## ΜΟΝΙΜΗ ΚΑΤΑΧΩΡΙΣΗ ΜΝΗΜΗΣ

- κανενός είδους πολιτική
- μόνιμη καταχώριση των περιοχών της μνήμης στις διεργασίες

## Συστήματα ειδικού σκοπού (special-purpose systems) και συστήματα πραγματικού χρόνου (real-time systems)

- όλες οι διεργασίες
- (αν) το σύστημα διαθέτει αρκετή μνήμη

## Συστήματα γενικού σκοπού

- πυρήνας και διεργασίες του συστήματος
- ενταμιευτές εισόδου/εξόδου
- ενταμιευτές μηνυμάτων
- πίνακες του συστήματος
  - αλλιώς: μεγάλη σπατάλη χρόνου
- από τη διαδικασία αρχικής φόρτωσης/εκκίνησης (bootstrap)

# Λογισμικό διαχείρισης μνήμης

## ΠΡΟΣΩΡΙΝΗ ΚΑΤΑΧΩΡΙΣΗ ΜΝΗΜΗΣ

### Συστήματα γενικού σκοπού

- όλες οι υπόλοιπες διεργασίες
- δύο φιλοσοφίες καταχώρισης της μνήμης
  - καταχώριση ορισμένου μέγιστου χώρου στις διεργασίες-γονείς
    - οι διεργασίες-γονείς καταχωρούν ένα υποσύνολο του χώρου τους στις διεργασίες-παιδιά
  - καταχώριση της μνήμης υπό τον έλεγχο του συστήματος
- σ' οποιαδήποτε διεργασία ανεξάρτητα από την ιεραρχία της

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΠΡΟΣΚΟΜΙΣΗΣ

- **ΠΟΤΕ ΚΑΙ ΠΟΙΑ ΤΕΜΑΧΙΑ** πρέπει να μεταφερθούν από τη ΒΜ στην ΚΜ?

## ΠΟΛΙΤΙΚΕΣ ΠΡΟΒΛΕΨΗΣ

- προτού χρειαστεί να προσπελαστούν
- όταν συμπεριληφθεί μια διεργασία στο σύνολο των κατάλληλων (από το διαχειριστή της ΚΜΕ)
- ποια τεμάχια?
  - με βάση προβλέψεις για τη μελλοντική συμπεριφορά του προγράμματος της διεργασίας

## ΠΟΛΙΤΙΚΕΣ ΑΙΤΗΣΗΣ

- το ζητούμενο τεμάχιο όταν συμβεί διακοπή του τεμαχίου αυτού
- ευκολότερη υλοποίηση

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΠΡΟΣΚΟΜΙΣΗΣ

### Σύγκριση πολιτικών Πρόβλεψης - Αίτησης

- **Πόροι που καταναλώνονται στις δύο περιπτώσεις**
  - Αίτηση: χρόνος εκτέλεσης διαδικασίας εξυπηρέτησης διακοπής
  - Πρόβλεψη: χρόνος επιλογής τεμαχίου που θα προσκομιστεί προκαταβολικά.
    - χρόνος ελεγκτή ή διαύλου που το μεταφέρει
    - χώρος της κύριας μνήμης που καταλαμβάνει το τεμάχιο
- Αν το τεμάχιο που θα μεταφερθεί προκαταβολικά δεν προσπελαστεί καθόλου, τότε το σύστημα θα έχει σπαταλήσει άδικα τους πόρους του
- Κατανάλωση πόρων στην περίπτωση λανθασμένης επιλογής >> οικονομία πόρων στην περίπτωση της ορθής επιλογής

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΠΡΟΣΚΟΜΙΣΗΣ

### Σύγκριση πολιτικών Πρόβλεψης - Αίτησης

- Έστω ότι επιλέγονται προκαταβολικά  $s$  τεμάχια μιας διεργασίας
- έστω  $\alpha$  το κλάσμα των τεμαχίων αυτών προσπελάζονται πριν αντικατασταθούν ( $0 \leq \alpha \leq 1$ )
- $A$  = κόστος επιβάρυνσης σφαλμάτων που εξοικονομήθηκε από την προκαταβολική προσκόμιση των  $\alpha \cdot s$  τεμαχίων
- $B$  = κόστος που ξοδεύτηκε άδικα για να προσκομιστούν τα  $(1-\alpha) \cdot s$  τεμάχια
- Αν  $A > B$ , τότε
  - είναι συμφέρουσα η πολιτική πρόβλεψης
  - μεγάλη πιθανότητα τα τεμάχια που προσκομίστηκαν προκαταβολικά να προσπελαστούν προτού ν' αντικατασταθούν

# Λογισμικό διαχείρισης μνήμης

## Σύγκριση πολιτικών Πρόβλεψης - Αίτησης

### ΑΛΛΑ

- οι μελλοντικές αναφορές ενός προγράμματος είναι πολύ δύσκολο να προβλεφτούν και
- (υπό ορισμένες προϋποθέσεις) μπορεί ν' αποδειχθεί ότι:

Για κάθε αλγόριθμο πρόβλεψης είναι δυνατό να υλοποιηθεί ένας αλγόριθμος αίτησης, που δεν προκαλεί περισσότερα σφάλματα από τον πρώτο για την κάθε ακολουθία αναφοράς ενός προγράμματος

- ακολουθία αναφοράς προγράμματος (program reference string) είναι η ακολουθία των τεμαχίων της διεργασίας που προσπελάζονται

### ΑΡΑ

- οι πολιτικές πρόβλεψης σπανίως χρησιμοποιούνται

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΤΟΠΟΘΕΤΗΣΗΣ

- Σε ΠΟΙΕΣ θέσεις της ΚΜ θα τοποθετηθούν τα τεμάχια που προσκομίζονται
- **Σελιδοποιημένα συστήματα**
  - Δε χρειάζονται πολιτικές τοποθέτησης γιατί όλα τα πλαίσια της μνήμης είναι ισομεγέθη
- **Τεμαχισμένα συστήματα**
  - Εύρεση ελεύθερης περιοχής μνήμης μεγαλύτερης ή ίσης με το μέγεθος του τεμαχίου που πρόκειται να προσκομιστεί
  - Αν το τεμάχιο τοποθετηθεί στην ΚΜ:
    - Λίστα κατειλημμένων : προστίθεται ένα στοιχείο
    - Λίστα ελεύθερων:
      - Μένει αμετάβλητο το πλήθος των στοιχείων (πότε?)
      - Προστίθεται ένα στοιχείο (πότε?)
        - Δε συμβαίνει στην πράξη (γιατί?)
      - Αφαιρείται ένα στοιχείο (πότε?)
        - Σπάνια (γιατί?)
  - έξι αλγόριθμοι τοποθέτησης

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΤΟΠΟΘΕΤΗΣΗΣ

### 1. Καλύτερο Ταίριασμα (Best Fit, BF)

- διαλέγει τη μικρότερη ελεύθερη περιοχή, η οποία χωράει το προσκομιζόμενο τεμάχιο
- προσπαθεί να ελαχιστοποιήσει την απώλεια χώρου
- λίστα ελευθέρων
  - ταξινομημένη με τη διεύθυνση βάσης των ελεύθερων περιοχών
    - μπορεί να χρειαστεί να ψαχτεί ολόκληρη η λίστα
  - ταξινομημένη με το μέγεθος των ελεύθερων περιοχών
    - πιθανώς να μην ψαχτεί ολόκληρη η λίστα
    - όμως χρονοβόρα διατήρηση της λίστας
- η εξέταση της λίστας είναι χρονοβόρα
- δημιουργία πολλών μικρών ελεύθερων περιοχών που
  - δεν μπορούν να χρησιμοποιηθούν από μελλοντικά τεμάχια

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΤΟΠΟΘΕΤΗΣΗΣ

### 2. Χειρότερο Ταίριασμα (Worst Fit, WF)

- διαλέγει τη μεγαλύτερη ελεύθερη περιοχή
- ιδέα: όση ελεύθερη περιοχή απομείνει θα είναι αρκετά μεγάλη για μελλοντικά τεμάχια
- η εξέταση της λίστας είναι χρονοβόρα
- οι μεγάλες σπές καταχωρούνται πρώτα
  - δεν είναι δυνατή η φιλοξενία άλλων μεγάλων τεμαχίων

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΤΟΠΟΘΕΤΗΣΗΣ

### 3. Πρώτο Ταίριασμα (First Fit, FF)

- ο αλγόριθμος αυτός διαλέγει την πρώτη (με τη σειρά διεύθυνσης) οπή που χωράει το τεμάχιο που πρόκειται να προσκομισθεί
- δε χρειάζεται να ψάξει ολόκληρη τη λίστα ελευθέρων
  - κατά μέσο όρο χρειάζεται να ψάξει μόνο τη μισή λίστα
  - λιγότερος χρόνος επιβάρυνσης από τους BF και WF
- απομένουν ελεύθερες περιοχές σχετικά μεγάλου μεγέθους
- πειράματα προσομοίωσης [Knuth]
  - στις περισσότερες περιπτώσεις ο εξωτερικός κατακερματισμός  $FF < BF$
- πολλές μικρές οπές στην αρχή της λίστας ελευθέρων
  - ο FF ξεκινάει πάντα από την αρχή της λίστας ελευθέρων
  - σταδιακά αυξάνεται ο χρόνος εύρεσης της κατάλληλης οπής

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΤΟΠΟΘΕΤΗΣΗΣ

### 4. Κυκλικό Πρώτο Ταίριασμα (Cyclic First Fit) ή Επόμενο Ταίριασμα (Next Fit, NF)

- αποφεύγει το μειονέκτημα του FF
- Η λίστα ελευθέρων είναι οργανωμένη κυκλικά
- κάθε φορά το ψάξιμο αρχίζει από το στοιχείο στο οποίο σταμάτησε την προηγούμενη φορά (ή από το επόμενο του)
- κατανέμει τον εξωτερικό κατακερματισμό ομοιόμορφα σ' ολόκληρη μνήμη
- ελαττώνει το χρόνο ψαξίματος της λίστας ελευθέρων
  - όχι πάντα ...

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΤΟΠΟΘΕΤΗΣΗΣ

### 5. Γρήγορο Ταίριασμα (Quick Fit, QF)

- διαφορετικές λίστες για τα πιο συνηθισμένα μεγέθη οπών
- πίνακας (σταθερού μεγέθους)
  - στοιχείο πίνακα
    - δείκτης της κεφαλής μιας λίστας οπών ίσου μεγέθους (π.χ. 4K, 8K, 12K κ.ο.κ.)
- εξαιρετικά γρήγορη εύρεση κατάλληλης οπής
- QF (και όλες οι μέθοδοι που χρησιμοποιούν λίστες ταξινομημένες με το μεγέθους των οπών)
  - **Ελευθέρωση περιοχής**
    - πρέπει να βρεθούν οι γειτονικές της οπές και να συγχωνευτούν μ' αυτήν
    - ψάξιμο σε (σχεδόν) όλες τις λίστες ελεύθερων
    - πολύ μεγάλος χρόνος επιβάρυνσης
  - αν, πάλι, δε γίνει συγχώνευση των οπών
    - η μνήμη σύντομα θα θρυμματιστεί σε πολλές μικρές άχρηστες οπές

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΤΟΠΟΘΕΤΗΣΗΣ

### 6. Σύστημα Φίλων (Buddy System, BS)

- Αν το μέγεθος της μνήμης είναι ίσο με  $2^s$
- Ο αλγόριθμος διαιρεί σταδιακά τη μνήμη σε ομάδες/περιοχές (blocks/regions) μεγέθους  $2^k$ ,  $k = 0, 1, \dots, s$
- Διατηρεί  $s+1$  λίστες ελεύθερων περιοχών
- Λίστα  $k$ : περιέχει ελεύθερες περιοχές μεγέθους  $2^k$
- Αρχικά
  - Όλη η μνήμη είναι μια οπή
  - Η λίστα  $s$  περιέχει την διεύθυνση της μιας οπής (όλη η μνήμη =  $2^s$ )
  - Οι λίστες  $0, 1, 2, \dots, s-1$  είναι κενές
- Μία οπή  $2^{k+1}$  μπορεί ν' αφαιρεθεί από τη λίστα  $k+1$  και να
  - Δημιουργηθεί ένα ζεύγος "φίλων" μεγέθους  $2^k$
  - Οι δυο φίλοι τοποθετούνται στη λίστα  $k$
- Ένα ζεύγος φίλων  $2^k$  μπορεί ν' αφαιρεθεί από τη λίστα  $k$  και να
  - Συγχωνευθεί
  - Η οπή  $2^{k+1}$  τοποθετείται στη λίστα  $k+1$
- Αναδρομικός αλγόριθμος εύρεσης της κατάλληλης οπής



# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΤΟΠΟΘΕΤΗΣΗΣ

### Σύστημα Φίλων (Buddy System - BS)

Παράδειγμα (Μνήμη 1M)

	0	128k	256k	512k	1024k	
start	1024k					
A=70K	A	128	256	512		
B=35K	A	B	64	256	512	
C=80K	A	B	64	C	128	512
A ends	128	B	64	C	128	512
D=60K	128	B	D	C	128	512
B ends	128	64	D	C	128	512
D ends	256		C	128	512	
C ends	512			512		
end	1024k					

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΤΟΠΟΘΕΤΗΣΗΣ

### Σύστημα Φίλων (Buddy System, BS)

- Όταν ελευθερώνεται μία περιοχή της μνήμης
  - η οπή που σχηματίζεται
    - είτε συνδέεται στην κατάλληλη λίστα
    - είτε, αν βρεθεί ότι συνορεύει με ένα φίλο της,
      - το ζεύγος αφαιρείται από τη λίστα
      - συγχωνεύεται
      - προστίθεται στην κατάλληλη λίστα
      - μόνο η λίστα των οπών μεγέθους  $2^k$  χρειάζεται να ψαχτεί
- Πειράματα προσομοίωσης [Knuth]
  - ταχύτητα: BS >(λίγο) FF
  - χώρος:
    - Απαιτήσεις διεργασιών ομοιόμορφα κατανομημένες στο  $[1, 2^s]$
    - μεγέθη περιοχών = δυνάμεις του 2
    - BS > 1.3 x FF
  - σοβαρός εσωτερικός θρυμματισμός

ΠΟΙΑ ΕΙΝΑΙ ΚΑΛΥΤΕΡΗ ΑΠΟ ΤΙΣ ΎΕΙ ΠΟΛΙΤΙΚΕΣ?

# Λογισμικό διαχείρισης μνήμης

## Συμπύκνωση (Compaction) ή αποκομιδή απορριμμάτων (garbage collection)

### Τεμαχισμός

- Προσκομιζόμενο τεμάχιο  $>$  μεγαλύτερη οπή
- Ένα θύμα δε δημιουργεί αρκετά μεγάλη οπή
- Αντικατάσταση περισσότερων από ένα τεμαχίων
  - μήκη των τμημάτων + μήκη των γειτονικών τους οπών ή τμημάτων  $>$  προσκομιζόμενο τεμάχιο
  - ή
  - συμπύκνωση της ελεύθερης μνήμης

### ΣΥΜΠΥΚΝΩΣΗ

- Μετακίνηση των κατειλημμένων περιοχών σ' ένα άκρο της μνήμης
- Σχηματισμός μίας μόνο οπής στο αντίθετο άκρο της μνήμης
- Ευκολότερη όταν ή λίστα ελεύθερων είναι ταξινομημένη σύμφωνα με τις διευθύνσεις βάσης των οπών

# Λογισμικό διαχείρισης μνήμης

## Συμπύκνωση ή αποκομιδή απορριμμάτων

### ΚΑΤΑΣΤΑΣΕΙΣ ΣΥΣΤΗΜΑΤΟΣ ΜΝΗΜΗΣ [Knuth]

- Σύστημα σε ισορροπία (equilibrium)

A) ο χώρος που ελευθερώνεται σε κάποιο χρονικό διάστημα είναι περίπου ίσος με το χώρο που καταχωρείται

B) **Κανόνας του 50%:** κατειλημμένες περιοχές = 2 x (ελεύθερες περιοχές)

Γ) οι ελεύθερες περιοχές είναι σχετικά μικρού μεγέθους  
ΑΡΑ

- μνήμη είναι σχεδόν γεμάτη
- η συμπύκνωση έχει μικρό όφελος

- Σύστημα μνήμης με απαιτήσεις μεγάλων τμημάτων

- έντονο πρόβλημα εξωτερικού θρυμματισμού
- μεγάλο ποσοστό της μνήμης μένει αχρησιμοποίητο
- η συμπύκνωση μπορεί να είναι συμφέρουσα, γιατί
  - αν το προς αντικατάσταση τμήμα είναι λερωμένο θα
  - πρέπει να μεταφερθεί από την KM στη BM
  - μεγάλος χρόνος επιβάρυνσης

# Λογισμικό διαχείρισης μνήμης

## Συμπύκνωση ή αποκομιδή απορριμμάτων

### Συχνότητα εκτέλεσης συμπύκνωσης

- **Όταν ελευθερωθεί μία περιοχή μνήμης**
  - διατηρείται συνεχώς συμπυκνωμένη η μνήμη
  - αποφεύγεται εντελώς ο εξωτερικός κατακερματισμός
  - δεν εξετάζεται ολόκληρη η λίστα ελεύθερων
  - πολύ μεγάλος χρόνος επιβάρυνσης
    - συχνή καταχώριση και ελευθέρωση περιοχών
- **Όταν χρειάζεται**
  - μεγάλος βαθμός θρυμματισμού
  - αδυναμία προσκόμισης απαιτούμενου τεμαχίου ενώ υπάρχει συνολικά αρκετός ελεύθερος χώρος
  - εξετάζεται ολόκληρη η λίστα ελεύθερων
    - μεγάλος χρόνος επιβάρυνσης
- **Οποτεδήποτε η ΚΜΕ είναι αδρανής**
  - εκμετάλλευση του χρόνου αδράνειας του συστήματος

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

- **Τεμαχισμός:** μέγεθος ζητούμενου > μέγεθος μεγαλύτερης οπής
- **Σελιδοποίηση:** δεν υπάρχει ελεύθερη σελίδα
- **Αντικατάσταση**
  - αφαίρεση ενός τεμαχίου από τη μνήμη
  - το αφαιρούμενο τεμάχιο ονομάζεται *θύμα (victim)*- ΠΟΙΟ ΤΕΜΑΧΙΟ?
  - δημιουργία ελεύθερης περιοχής (που να χωράει το τεμάχιο)
  - τοποθέτηση ζητούμενου τεμαχίου
- **Λίστα κατειλημμένων:** αφαίρεση ενός στοιχείου
- **Λίστα ελευθέρων:**
  - πρόσθεση ενός στοιχείου (πότε?)
  - ίδιος αριθμός στοιχείων (πότε?)
  - αφαίρεση ενός στοιχείου (πότε?)

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### ΚΡΙΤΗΡΙΑ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ ΤΕΜΑΧΙΟΥ

- Χρόνος (ΚΜΕ ή πραγματικός) που έχει περάσει από την τελευταία προσπέλαση του τεμαχίου
- Συχνότητα προσπέλασης του τεμαχίου
- Συνολικό πλήθος τεμαχίων της διεργασίας στην κύρια μνήμη
- Είναι η διεργασία στην λίστα κατάλληλων (για διανομή στην ΚΜΕ)
- Προτεραιότητα ή ιεραρχία της διεργασίας σε σχέση με τη διεργασία του προσκομιζόμενου τεμαχίου
- (Κατα)μερισμός τεμαχίων
- κ.ά.

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

- Καταχώριση περιοχών του δίσκου στις διεργασίες (βλ. Κεφ. 10)
  - παρόμοια με την τοποθέτηση στην κύρια μνήμη

### Περιοχή Ανταλλαγής (Swap Area) μιας διεργασίας

- ειδικός χώρος στο δίσκο (σταθερής κεφαλής)
  - μοναδική συνεχόμενη περιοχή του δίσκου
- δημιουργία διεργασίας: καταχώριση περιοχής
- αποθήκευση “λερωμένων τεμαχίων”
- διαγραφή διεργασίας: αποθήκευση “λερωμένων” τεμαχίων στη ΒΜ και ελευθέρωση του χώρου που κατείχε στην περιοχή ανταλλαγής

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

**Χώρος Εργασίας (XE) διεργασίας (workspace)** είναι ο χώρος της κύριας μνήμης που διατίθεται για τα τεμάχιά της

### **Τοπικοί αλγόριθμοι (Local algorithms) αντικατάστασης**

- Σε κάθε διεργασία καταχωρείται ένας XE σταθερού μεγέθους
- Στατική καταχώριση
- Οι αλγόριθμοι αντικατάστασης εφαρμόζονται ανεξάρτητα στον κάθε XE
- *Ένα σφάλμα τεμαχίου μιας διεργασίας μπορεί να αντικαταστήσει ένα τεμάχιο μόνο από το δικό της XE*

### **Καθολικοί/σφαιρικοί αλγόριθμοι αντικατάστασης (Global replacement algorithms)**

- Το μέγεθος του XE μιας διεργασίας μεταβάλλεται δυναμικά
- Οι αλγόριθμοι εφαρμόζονται σε όλη τη μνήμη
- *Ένα σφάλμα τεμαχίου μιας διεργασίας μπορεί ν' αντικαταστήσει τεμάχιο οποιασδήποτε άλλης διεργασίας*

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

Ακολουθία αναφοράς προγράμματος (program reference string) είναι η ακολουθία των τεμαχίων της διεργασίας που προσπελάζονται, π.χ.  $s_1, s_2, s_3, \dots, s_{k-1}, s_k, s_{k+1}, s_{k+2}$

### Σύνολο Εργασίας (Working Set, $W$ ) διεργασίας [Denning, 1968]

$W(k)$  = σύνολο των τεμαχίων που προσπελάζονται μέχρι την  $k$  αναφορά της ( $k \geq 1$ ),

π.χ.  $W(k) = s_1, s_2, s_3, \dots, s_{k-1}, s_k$

ή

$W(t, \Delta t)$  = σύνολο των τεμαχίων που προσπελάζονται στο χρονικό διάστημα  $(t-\Delta t, t)$ ,

δηλαδή το σύνολο των τεμαχίων που έχουν προσπελαστεί πρόσφατα (στο τελευταίο χρονικό διάστημα  $\Delta t$ )

- $\Delta t$  = παράθυρο χρόνου (window of time)
- σειρά συνόλων εργασίας για μία διεργασία
- εξαρτώμενα από
  - τη χρονική στιγμή  $t$  και
  - το παράθυρο χρόνου  $\Delta t$  που μετριούνται
- μέγεθος συνόλου εργασίας  $w(t, \Delta t) = |W(t, \Delta t)|$
- $\Delta t=0 \Rightarrow W(t, 0) = \{\}$  και  $w(t, 0)=0$
- $w(t, \Delta t)$ : μη φθίνουσα συνάρτηση του  $\Delta t$
- $w(t, \Delta t) \leq N$  όπου  $N$ = πλήθος των τεμαχίων που χρησιμοποιεί η διεργασία κατά την εκτέλεσή της
- $N \leq$  συνολικό πλήθος των τεμαχίων της διεργασίας (γιατί ?)

- W: χρήσιμο για την μελέτη χρήσης της μνήμης

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### ΑΛΩΝΙΣΜΑ

**Μία (καμία) διεργασία δεν μπορεί να διατηρήσει το σύνολο εργασίας της στη μνήμη, δηλαδή δεν μπορεί να διατηρήσει αρκετά τεμάχιά της στη μνήμη για να τα εκτελέσει για κάποιο λογικό χρονικό διάστημα**

Ποιες πολιτικές ευνοούν το αλώνισμα?

#### Καθολικές πολιτικές [Denning]

- Δεν υπάρχει τρόπος προσδιορισμού του συνωστισμού (congestion) της μνήμης
- Ο “χώρος εργασίας” της διεργασίας μπορεί να μη χωράει το σύνολο εργασίας της
- Αντικατάσταση τεμαχίων άλλων διεργασιών που αυτές μπορεί να χρειαστούν σύντομα
- Αντικατάσταση δικών της τεμαχίων που μπορεί αυτή να χρειαστούν σύντομα
- Ο χρόνος εκτέλεσης μιας διεργασίας διαφέρει σε κάθε της εκτέλεση, γιατί ο χώρος εργασίας της διεργασίας εξαρτάται από τη δραστηριότητα των άλλων διεργασιών που βρίσκονται στο σύστημα ταυτόχρονα με αυτή

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### ΑΛΩΝΙΣΜΑ

#### Τοπικές πολιτικές [άλλοι επιστήμονες]

- Μικρός χώρος εργασίας => αλώνισμα
  - Παρόλο που πιθανώς υπάρχει ελεύθερη μνήμη
- Μεγάλος χώρος εργασίας => άδικη σπατάλη μνήμης. Άλλες διεργασίες (μπορεί να) τον χρειάζονται

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

**ΣΤΟΧΟΣ** πολιτικών αντικατάστασης (καθολικών και τοπικών)

- Αποφυγή αλωνίσματος
- Ελαχιστοποίηση της κυκλοφορίας τεμαχίων, δηλαδή ελαχιστοποίηση του πλήθους των σφαλμάτων
  - Επιλογή τεμαχίων που δεν είναι χρήσιμα
    - αυτά που έχουν την μικρότερη πιθανότητα να προσπελαστούν στο άμεσο μέλλον

**Ανεπίσημη αρχή της βελτιστότητας (informal principle of optimality) [Belady]**

Ένας αλγόριθμος που επιλέγει το τεμάχιο που δε θα προσπελαστεί για το μεγαλύτερο μελλοντικό χρονικό διάστημα, δημιουργεί το μικρότερο πλήθος σφαλμάτων –σε σχέση με το πλήθος των αναφορών– από οποιονδήποτε άλλον αλγόριθμο

- **Αλγόριθμος MIN ή OPT [Belady]**
  - χρειάζεται να γνωρίζει ακριβώς τη μελλοντική συμπεριφορά των διεργασιών
  - δεν είναι δυνατό να υλοποιηθεί
  - χρήσιμος για τη σύγκριση άλλων αλγορίθμων

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Τυχαία επιλογή

- επιλέγει ένα τεμάχιο στην τύχη
- πολύ απλή υλοποίηση
- μπορεί, όμως, να αντικαταστήσει χρήσιμα τεμάχια

### Πρώτο-Μέσα-Πρώτο-Έξω (FIFO) ή Γηραιότερο διαμένον (Oldest resident)

- επιλέγει το γηραιότερο τεμάχιο, δηλαδή αυτό που βρίσκεται στη μνήμη τον περισσότερο χρόνο
- εύκολη και οικονομική υλοποίηση (ουρά)
- *πιθανότητα αντικατάστασης χρήσιμων τεμαχίων*  
(το γηραιότερο τεμάχιο δεν είναι πάντα το πιο άχρηστο)
  - ειδικά όταν το σύστημα έχει μεγάλο φόρτο εργασίας
  - οι αναφορές στα δεδομένα δεν είναι πάντα ακολουθιακές
    - αν και οι εντολές μιας διεργασίας εκτελούνται συνήθως διαδοχικά (αν αγνοήσουμε τις ανακυκλώσεις)
- *υποφέρει από την “ανωμαλία” του Belady (Belady’s anomaly)*
  - αύξηση της συχνότητας των σφαλμάτων, όταν αυξάνονται τα διαθέσιμα πλαίσια (ή οι χώροι εργασίας) της μνήμης!!! (βλ. Άσκ. και Απάν. 8.1.5)

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Λιγότερο Πρόσφατα Χρησιμοποιημένο (Least Recently Used, LRU)

- επιλέγει το τεμάχιο που δεν έχει προσπελαστεί για το μεγαλύτερο χρονικό διάστημα
- υπόθεση: μελλοντική συμπεριφορά διεργασίας θα είναι παρόμοια της πρόσφατης συμπεριφοράς της
- δημοφιλής πολιτική

### Προβλήματα

- Είναι πιθανό στο εγγύς μέλλον να προσπελαστεί κάποιο τεμάχιο παρόλο που δεν έχει προσπελαστεί πρόσφατα
- Επιβάρυνση χρόνου για την καταγραφή των ακολουθιών των αναφορών στα τεμάχια
- Αντικατάσταση τεμαχίων επειδή η διεργασία δεν έχει πρόσφατα διανεμηθεί
  - Χρονοπρογραμματισμός RR
  - Καθολική πολιτική LRU
  - Η διεργασία που πρόκειται να διανεμηθεί είναι αυτή με τα λιγότερα τεμάχια στη μνήμη!

Το αλώνισμα που οφείλεται σε συνδυασμό αλγορίθμων αντικατάστασης με αλγόριθμους χρονοπρογραμματισμού της ΚΜΕ

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### LRU: Υλοποίηση με λογισμικό

- λίστα τεμαχίων που βρίσκονται στη μνήμη
- ταξινομημένη με το χρόνο από την τελευταία αναφορά
- ενημέρωση σε κάθε αναφορά στη μνήμη
- δημιουργία παγίδας για ενημέρωση
  - υπερβολική καθυστέρηση προσπέλασης της μνήμης!!!

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### LRU: Υλοποίηση με υλισμικό

#### Μετρητές χρόνου

- ειδικός καταχωρητής-μετρητής (π.χ. 64 bits)
  - αυξάνεται αυτόματα κατά 1, κάθε φορά που γίνεται αναφορά στη μνήμη (υλισμικό)
- “πεδίο χρόνου” στους περιγραφητές τεμαχίων
  - αποθηκεύει την τιμή του καταχωρητή-μετρητή, κάθε φορά που γίνεται αναφορά στο τεμάχιο αυτό
  - παριστάνει έτσι το “χρόνο” της τελευταίας αναφοράς στο τεμάχιο
- αντικατάσταση: ψάχνονται όλοι οι περιγραφητές για να βρεθεί το τεμάχιο που έχει το μικρότερο πεδίο χρόνου
- υπερχείλιση του καταχωρητή-μετρητή
  - αντιμετωπίζεται από το διαχειριστή μνήμης

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### LRU: Υλοποίηση με υλισμικό

#### Μήτρα χρόνου (σελιδοποίηση)

- Μνήμη  $n$  πλαισίων σελίδων
- Μήτρα δυφίων  $n \times n$
- Αρχικά μηδενική
- Κάθε φορά που γίνεται αναφορά σ' ένα πλαίσιο,  $f$ , το υλισμικό:
  - κάνει όλα τα στοιχεία της γραμμής  $f$  ίσα με 1
  - κάνει όλα τα στοιχεία της στήλης  $f$  ίσα με 0
- Αντικατάσταση σελίδας
  - επιλέγεται ή γραμμή που έχει τη μικρότερη δυαδική τιμή, δηλαδή το πλαίσιο που χρησιμοποιήθηκε λιγότερο πρόσφατα

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### LRU: Υλοποίηση με συνδυασμό υλισμικού και λογισμικού

#### Καταχωρητές ηλικίωσης (Ageing Registers, ARs)

AR = μετρητής σε κάθε περιγραφητή τεμαχίου

- Σε κάθε αναφορά σε τεμάχιο:  $AR \leftarrow$  αρχική τιμή
- Περιοδικά (διάστημα ηλικίωσης):  
 $AR \leftarrow AR - 1$  ή  $AR \leftarrow \text{Shift Right}1 (AR)$
- η ελάττωση των περιεχομένων ενός AR αυξάνει την ηλικία του αντίστοιχου τεμαχίου
- Το τεμάχιο που έχει τη μικρότερη τιμή στο μετρητή του είναι αυτό που έχει λιγότερο πρόσφατα χρησιμοποιηθεί
- **Προβλήματα**
  - Όλα τα τεμάχια, που οι μετρητές τους έχουν τα ίδια περιεχόμενα θεωρούνται ότι έχουν την ίδια ηλικία
  - Η ελάττωση των μετρητών είναι χρονοβόρα
    - Ειδικά αν το διάστημα ηλικίωσης είναι μικρό
  - Η αναζήτηση του μετρητή με το μικρότερο περιεχόμενο είναι χρονοβόρα
    - Κατάλληλο διάστημα ηλικίωσης
    - Κατάλληλο μήκος μετρητών

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### LRU: Υλοποίηση με συνδυασμό υλισμικού και λογισμικού

#### Στοίβα

- περιέχει τους αριθμούς των τεμαχίων που έχουν πρόσφατα προσπελαστεί
- Σε κάθε αναφορά σε τεμάχιο:
  - ο αριθμός του αφαιρείται από τη στοίβα (αν υπάρχει σ' αυτή) και
  - προστίθεται στην κορυφή της
- Κρυφή στοίβας: αριθμός πιο πρόσφατα χρησιμοποιημένου τεμαχίου
- Βάση στοίβας: αριθμός λιγότερο πρόσφατα χρησιμοποιημένου
- Αφαίρεση αριθμών τεμαχίων από ενδιάμεσες θέσεις της στοίβας
- Υλοποίηση στοίβας με γραμμική λίστα δύο διευθύνσεων
- Αφαίρεση στοιχείου στοίβας: αλλαγή έξι δεικτών (το πολύ)
- Χρονοβόρα ενημέρωση στοίβας
- Εύκολη επιλογή τεμαχίου
  - ο δείκτης της ουράς της δείχνει στο τεμάχιο που έχει λιγότερο πρόσφατα χρησιμοποιηθεί
- κατάλληλη μέθοδος για υλοποίηση με μικροπρογραμματισμό

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Λιγότερο Συχνά Χρησιμοποιημένο (Least Frequently Used, LFU)

- επιλέγει το τεμάχιο που έχει χρησιμοποιηθεί λιγότερο συχνά κατά τη διάρκεια κάποιου προηγούμενου χρονικού διαστήματος
- ίδια υπόθεση με LRU
- παρόμοια μειονεκτήματα με LRU
- τεμάχια που έχουν πρόσφατα φορτωθεί στη μνήμη έχουν χρησιμοποιηθεί λίγο και είναι πιθανό να αντικατασταθούν
  - λύση: να μην επιλέγονται τα τεμάχια που έχουν φορτωθεί στο πιο πρόσφατο χρονικό διάστημα

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Λιγότερο Συχνά Χρησιμοποιημένο (Least Frequently Used, LFU)

#### Υλοποίηση

- για κάθε τεμάχιο: Μετρητής Χρήσης ή Αναφοράς (Use ή Reference Count, UC)
- φόρτωση τεμαχίου στη μνήμη:  $UC = 0$ 
  - αναφορά στο τεμάχιο :  $UC = UC + 1$

ή

- αναφορά στο τεμάχιο: δυφίο χρήσης (στον περιγραφητή) τεμαχίου = 1
- *περιοδικά*: τα δυφία χρήσης (των περιγραφητών τεμαχίων)
  - προστίθενται στους αντίστοιχους UCs
  - και μηδενίζονται
- επιλέγεται το τεμάχιο με τον μικρότερο UC ( $\neq 0$ )

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Λιγότερο Συχνά Χρησιμοποιημένο (Least Frequently Used, LFU)

#### Μειονέκτημα:

- τεμάχια που είχαν χρησιμοποιηθεί συχνά στο παρελθόν και που δε χρειάζονται πλέον μπορεί να εξακολουθούν να έχουν μεγαλύτερους μετρητές χρήσης από τεμάχια που έχουν πρόσφατα φορτωθεί!
  - **Λύση:** *περιοδικά* οι UCs
    - Μετατοπίζονται μια θέση δεξιά, και μετά
    - το δυφίο χρήσης προστίθεται στο πσδ αντί στο λσδ
    - τα τεμάχια που δεν έχουν χρησιμοποιηθεί πρόσφατα έχουν μικρότερους μετρητές χρήσης
- Απλούστευση/επιτάχυνση υλοποίησης με τη βοήθεια υλισμικού

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Δυφίο Χρήσης ή Αναφοράς (Use ή Reference Bit)

- LRU, καταχωρητής ηλικίωσης, AR
- LFU, μετρητής χρήσης, UC
- LRU= LFU=FIFO με ARs ή UCs μήκους 0
  - τα τεμάχια με ίδια ηλικία ή την ίδια συχνότητα αναφοράς επιλέγονται με τη σειρά FIFO
- δυφίο χρήσης ή αναφοράς = μετρητής μήκους 1
  - γίνεται 1 όταν γίνει αναφορά στο τεμάχιο

### Μειονέκτημα

- το δυφίο χρήσης δε δίνει την πραγματική εικόνα της δραστηριότητας ενός τεμαχίου
- δυφίο χρήσης ==1
  - 1 ή 1000000 προσπελάσεις?
  - Τελευταία προσπέλαση πριν 1ms ή 1min?
  - όλα τα τεμάχια με δυφία χρήσης ίσα με 1 θεωρούνται εξίσου δραστήρια
  - είναι πιθανό να αντικατασταθούν χρήσιμα τεμάχια

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Δυφίο Χρήσης ή Αναφοράς (Use ή Reference Bit)

*Επίδοση συστήματος και μήκος μετρητών [Corbrató]*

- αύξηση μήκους μετρητών από 0 σε 1: σημαντική διαφορά
- αύξηση μήκους μετρητών από 1 σε  $k > 1$ : μικρή διαφορά

### Η πιο δημοφιλής πολιτική αντικατάστασης

- Εμπειρική διαπίστωση: η επίδοση της είναι σχετικά καλή στις περισσότερες περιπτώσεις
- Προσεγγίζει καλά τις LRU και LFU, όταν χρησιμοποιείται μικρό διάστημα χρόνου
- Είναι εύκολο να υλοποιηθεί στο υλισμικό και στο λογισμικό

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Αλγόριθμος Δεύτερης Ευκαιρίας (Second Chance Algorithm)

- Συνδυασμός FIFO με την πολιτικής δυφίου χρήσης
- Διατήρηση *κυκλικής λίστας* ταξινομημένης με βάση το χρόνο παραμονής στη μνήμη (FIFO)
- Εξετάζεται το δυφίο χρήσης του επόμενου τεμαχίου
- Αν δυφίο χρήσης = 1, τότε δυφίο χρήσης = 0 (του δίνεται δεύτερη ευκαιρία) και
  - Εξετάζεται το επόμενο στοιχείο της λίστας
- Αν δυφίο χρήσης = 0
  - Το τεμάχιο επιλέγεται ως θύμα

Ένα τεμάχιο στο οποίο δίνεται μια δεύτερη ευκαιρία δεν αντικαθίσταται, μέχρις ότου όλα τα υπόλοιπα τεμάχια είτε

- αντικατασταθούν είτε
- τους δοθεί μια δεύτερη ευκαιρία

Αν ένα τεμάχιο χρησιμοποιείται συχνά

- μπορεί να μην αντικατασταθεί καθόλου κατά τη διάρκεια της εκτέλεσης της διεργασίας του

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Αλγόριθμος Δεύτερης Ευκαιρίας (Second Chance Algorithm)

- Χειρότερη περίπτωση
  - όλα τα δυφία χρήσης είναι ίσα με 1
  - εκτελείται ένας πλήρης κύκλος της λίστας
  - κάθε τεμάχιο παίρνει μια δεύτερη ευκαιρία, άρα FIFO
- Εύκολη υλοποίηση
- Διατηρεί στη μνήμη τα χρήσιμα τεμάχια και ανταλλάσσει τα άχρηστα
- Προσεγγίζει καλά τις LRU και LFU, όταν το διάστημα εξέτασης της λίστας είναι μικρό
- **Μειονεκτήματα** παρόμοια με την πολιτική του δυφίου χρήσης

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Προτεραιότητες Τεμαχίων

Από τους προγραμματιστές/μεταφραστές

- **Ανεπίσημη αρχή της τοπικότητας (Informal principle of locality)**

**Κατά τη διάρκεια ενός χρονικού διαστήματος εκτέλεσης, μία διεργασία τείνει να χρησιμοποιεί ένα υποσύνολο των τεμαχίων της και ότι τα μέλη του υποσυνόλου αυτού αλλάζουν σχετικά αργά**

- Αν η γλώσσα υψηλού επιπέδου είναι καλά δομημένη
  - Ο προγραμματιστής/μεταφραστής μπορεί να βοηθήσει το σύστημα να επιλέξει τα τεμάχια που έχουν μικρή πιθανότητα χρήσης στο εγγύς μέλλον
- Στην πράξη μόνο σε μεμονωμένες περιπτώσεις

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Ανάλογα με τα δυφία χρήσης και “ακαθαρσίας”

- Διανομή διεργασίας : τα δυφία χρήσης (used) και ακαθαρσίας (dirty) όλων των τεμαχίων της γίνονται μηδέν
- Περιοδικά : τα δυφία χρήσης όλων των τεμαχίων της γίνονται μηδέν (για τα τεμάχια που έχουν προσπελαστεί πριν από το χρόνο αυτό)
- Τάξεις τεμαχίων (για το τελευταίο διάστημα)
  - Τάξη 0 (0, 0): δεν έχει χρησιμοποιηθεί και είναι καθαρό
  - Τάξη 1 (0, 1): δεν έχει χρησιμοποιηθεί και είναι λερωμένο
  - Τάξη 2 (1, 0): έχει χρησιμοποιηθεί και είναι και είναι καθαρό
  - Τάξη 3 (1, 1): έχει χρησιμοποιηθεί και είναι λερωμένο
- Επιλέγονται τα τεμάχια της μικρότερης (μη κενής) τάξης
  - Αν υπάρχουν πολλά τεμάχια στην τάξη αυτή
    - τυχαία ή FIFO

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Προτεραιότητες Διεργασιών

- Καθολική πολιτική
- Επιλέγει ένα ή περισσότερα από τα τεμάχια της διεργασίας με τη μικρότερη προτεραιότητα με κάποια πολιτική, π.χ. LRU
- Αν το σφάλμα τεμαχίου έχει δημιουργηθεί από τη διεργασία με τη μικρότερη προτεραιότητα
  - Μπορεί να επιλεγεί ένα από τα δικά της τεμάχια
  - Αν δεν υπάρχει κανένα τεμάχιο της στη μνήμη η διεργασία εμποδίζεται ή αναστέλλεται
- **Μειονέκτημα**
  - Τα τεμάχια που δεν απαιτούνται πια από διεργασίες με μεγάλη προτεραιότητα εξακολουθούν να παραμένουν στη μνήμη
    - Καμία διεργασία μικρότερης προτεραιότητας δεν μπορεί να τα κλέψει
- **Πλεονέκτημα**
  - Βασίζεται πραγματικά στη μελλοντική δραστηριότητα του συστήματος
  - Ο διαχειριστής της ΚΜΕ αποφασίζει ποιες διεργασίες πρόκειται να εκτελεστούν στο μέλλον

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Προτεραιότητες Διεργασιών

Συνδυασμός αλγορίθμων αντικατάστασης με αλγόριθμους χρονοπρογραμματισμού ΚΜΕ

- προσεγγίζει καλύτερα απ' όλες τις προηγούμενες την αρχή της βελτιστότητας

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Άλλες πολιτικές

- κανόνες κοινής λογικής εξίσου αποτελεσματικοί

### Δεξαμενή (pool) ελεύθερων πλαισίων

- Το σύστημα διατηρεί ένα ορισμένο πλήθος ελεύθερων πλαισίων
- Το θύμα επιλέγεται με μία από τις προηγούμενες πολιτικές
- Αλλά δεν αντικαθίσταται αμέσως
- Η ζητούμενη σελίδα φορτώνεται σε ένα ελεύθερο πλαίσιο της δεξαμενής
- Αργότερα μεταφέρεται το θύμα στην περιοχή ανταλλαγής
  - όταν η συσκευή ανταλλαγής είναι αδρανής
  - όταν δεν υπάρχουν αρκετά ελεύθερα πλαίσια και
- Το πλαίσιο του επιστρέφεται στη δεξαμενή των ελεύθερων πλαισίων

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Παραλλαγή δεξαμενής (pool) ελεύθερων πλαισίων (VAX/VMS)

- FIFO + δεξαμενή ελεύθερων πλαισίων
  - Η FIFO μπορεί να αντικαταστήσει χρήσιμες σελίδες
  - Τα περιεχόμενα ενός πλαισίου δεν αλλάζουν όταν μεταφέρονται στο δίσκο κι έτσι
  - Μια σελίδα μπορεί να χρησιμοποιηθεί αμέσως όταν απαιτηθεί, προτού το πλαίσιο καταχωρηθεί σε κάποια άλλη διεργασία
  - Το σύστημα “θυμάται” ποιες σελίδες βρίσκονται στα πλαίσια της δεξαμενής
- Όταν συμβεί σφάλμα σελίδας, η ζητούμενη σελίδα μπορεί ακόμα να βρίσκεται ακόμη στη μνήμη!

## ΆΛΛΕΣ ΠΟΛΙΤΙΚΕΣ

### Λίστα λερωμένων τεμαχίων

- Διατηρείται από το σύστημα
- Όταν η συσκευή ανταλλαγής είναι αδρανής
  - επιλέγεται ένα λερωμένο τεμάχιο και γράφεται το δίσκο
  - το δυφίο ακαθαρσίας του γίνεται ίσο με 0
- Αυξάνεται η πιθανότητα να μη χρειαστεί να γραφεί το τεμάχιο στο δίσκο, αν αργότερα επιλεγεί σα θύμα

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### ΆΛΛΕΣ ΠΟΛΙΤΙΚΕΣ

#### Σύνθετη καθολική πολιτική (υποθετικό σύστημα)

Επιλογή (κατά σειρά “χρησιμότητας”) τεμαχίων που ανήκουν σε μια διεργασία που:

1. περιμένει να διαγραφεί
  2. περιμένει για την αλληλεπίδραση του χρήστη από το τερματικό
  3. περιμένει να τελειώσει μια διεργασία-παιδί της
  4. έχει μικρότερη προτεραιότητα από τη διεργασία που έσφαλε (και κατά προτίμηση δεν ανήκει στο σύνολο των κατάλληλων διεργασιών)
  5. δεν ανήκουν στο σύνολο εργασίας της διεργασίας που έσφαλε
- Αν δεν επιλεγεί θύμα ούτε με τον κανόνα 5
    - **αλλάζει η κατάσταση της διεργασίας σε “έχει ανασταλεί” (για τεμάχιο του συνόλου εργασίας της)**

Για κάθε βήμα:

- Πολιτική δυφίου χρήσης
- Αν όλα τα δυφία χρήσης είναι ίσα με 1, τότε τυχαία επιλογή
- *Καλή προσέγγιση της αρχής της βελτιστότητας*

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### Τεμάχια εισόδου/εξόδου

**Πρόβλημα:** Έστω ότι

- μία διεργασία  $\delta$  απαιτεί  $\epsilon/\epsilon$  από/προς ένα ενταμιευτή  $B$
- η διεργασία εμποδίζεται
- ο έλεγχος μεταφέρεται στο διανομέα
- διανέμεται μία άλλη διεργασία  $\delta'$
- η  $\delta'$  δημιουργεί σφάλμα τεμαχίου
- επιλέγεται ως θύμα το τεμάχιο που περιέχει τον  $B$
- ο ελεγκτής της συσκευής  $\epsilon/\epsilon$  προσπελάζει άμεσα τη μνήμη (DMA) και λειτουργεί ταυτόχρονα με την αντικατάσταση των τεμαχίων τότε ένα μέρος της  $\epsilon/\epsilon$  θα γίνει από/προς τον  $B$  και το υπόλοιπο από/στο τεμάχιο που έχει προσκομιστεί!

**Λύσεις:**

- μεταφορές από/σε ενταμιευτές του πυρήνα
  - αντιγραφή στους/από τους ενταμιευτές  $\epsilon/\epsilon$  (χρονοβόρα λύση)
- κλείδωμα των τεμαχίων των διεργασιών που χρησιμοποιούνται για  $\epsilon/\epsilon$ 
  - τα κλειδωμένα τεμάχια δεν ανταλλάσσονται, είναι *απρόσβλητα από ανταλλαγή (immune from swapping)*
    - όταν ολοκληρωθεί η μεταφορά τα τεμάχια ξεκλειδώνονται

# Λογισμικό διαχείρισης μνήμης

## ΠΟΛΙΤΙΚΕΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ

### (Κατα)μεριζόμενα τεμάχια

- Δεν πρέπει να επιλέγονται σαν θύματα αν μια διεργασία που τα χρησιμοποιεί πάψει να υπάρχει, γιατί
- αν χρησιμοποιούνται και από άλλες διεργασίες θα αυξηθεί η συχνότητα των σφαλμάτων τους
- Λίστα-πίνακας (κατα)μεριζόμενης μνήμης (shared memory table)
  - περιέχει τις ταυτότητες των διεργασιών που χρησιμοποιούν τα (κατα)μεριζόμενα τεμάχια
  - για να μην ψάχνονται οι πίνακες τεμαχίων όλων των διεργασιών

# Λογισμικό διαχείρισης μνήμης

## UNIX (4.2BSD)

- **Τεμαχισμός με σελιδοποίηση**
  - σελίδες μήκους 1 Kb
- **Μόνιμα στη Μνήμη**
  - επόπτης (γνωστός ως πυρήνας του UNIX)
  - πίνακας διεργασιών
  - πίνακας κειμένων
  - χάρτης μνήμης (core map ή cmap)
    - στοιχεία μήκους 16 bytes
    - στοιχείο = περιγραφητής πλαισίου

# Λογισμικό διαχείρισης μνήμης

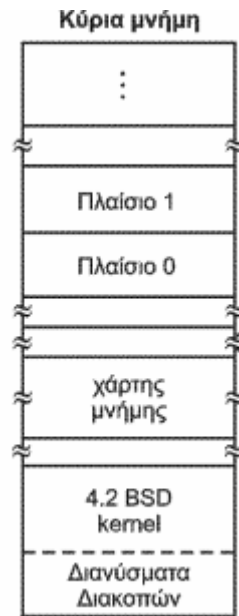
## UNIX (4.2BSD)

### Υπόλοιπη διαθέσιμη μνήμη

- πλαίσια σελίδων μήκους 1Kb
  - περιέχουν σελίδες ενός τμήματος
    - κειμένου
    - δεδομένων χρήστη
      - σωρός
      - στοίβα
    - δεδομένων συστήματος
      - δομή χρήστη (περιβάλλον)
      - στοίβα συστήματος
    - πίνακα σελίδων
  - βρίσκονται στη λίστα των ελεύθερων πλαισίων
    - λίστα δύο κατευθύνσεων

# Λογισμικό διαχείρισης μνήμης

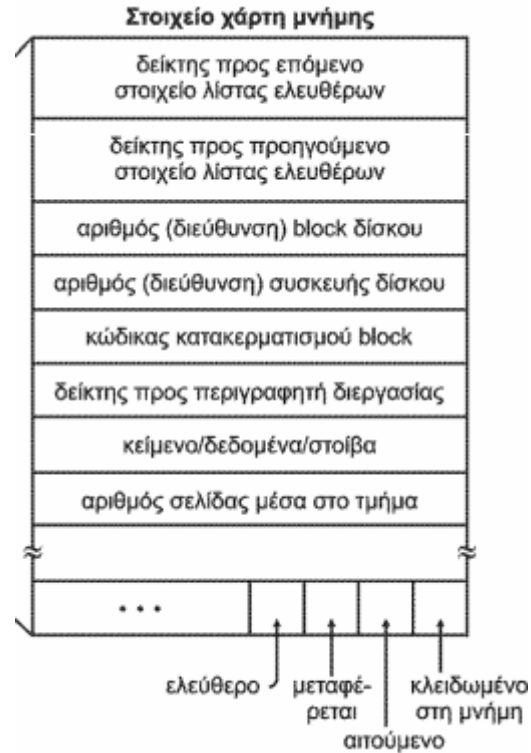
## UNIX (4.2BSD)



# Λογισμικό διαχείρισης μνήμης

## UNIX (4.2BSD)

- Στοιχείο χάρτη μνήμης = περιγραφητής πλαισίου



# Λογισμικό διαχείρισης μνήμης

## UNIX (4.2BSD)

### Πεδία περιγραφητών πλαισίων

- 1-2: (ελεύθερο) δείκτες προς τη λίστα των ελεύθερων πλαισίων
- 3-5: (κατειλημμένο) θέση της σελίδας στο δίσκο
- 6: θέση του περιγραφητή της διεργασίας στον πίνακα των διεργασιών
- 7: κατηγορία τμήματος στο οποίο ανήκει η σελίδα
- 8: αριθμός σελίδας στο τμήμα
- . . .
- άλλα δυφία που χρησιμοποιεί ο διαχειριστής μνήμης

# Λογισμικό διαχείρισης μνήμης

## UNIX (4.2BSD)

### ΠΕΡΙΟΧΗ ΑΝΤΑΛΛΑΓΗΣ

- καταχωρείται ανά περιοχές προκαθορισμένου (ίσου) πλήθους σελίδων
- $p$  = πλήθος των ελευθέρων πλαισίων της μνήμης
- πρέπει να ισχύουν οι σχέσεις  $lotsfree > desfree > minfree$
- $desfree$  και  $minfree$ 
  - ~ περιοχής δίσκου που καταχωρείται για ανταλλαγή
  - καθορίζονται κατά τη δημιουργία του συστήματος (system generation)

### ΠΡΟΣΚΟΜΙΣΗ

- *Στόχος: ελαχιστοποίηση πλήθους σφαλμάτων*
- **συνδυασμός**
  - προκαταβολικής προσκόμισης (προσελιδοποίησης) και
  - σελιδοποίησης με αίτηση
- **προσελιδοποίηση - πριν μια διεργασία αρχίσει την εκτέλεσή της**
  - πίνακας σελίδων
  - δομή χρήστη (περιβάλλοντος)
- **με αίτηση**
  - οι υπόλοιπες σελίδες

# Λογισμικό διαχείρισης μνήμης

## UNIX (4.2BSD)

### ΥΛΟΠΟΙΗΣΗ

- Διαχείριση μνήμης = διεργασία `swapper` (`pid=0`)
- (Διεργασία ρίζα = `init` (`pid=1`))
- Αλγόριθμος αντικατάστασης = διεργασία `pagedemon` (`pid=2`)

### ΠΟΛΙΤΙΚΗ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ (καθολική)

- εξετάζει κυκλικά τα πλαίσια σελίδων της μνήμης
- 1<sup>η</sup> πρώτη σάρωση: δυφίο χρήσης = 0
- 2<sup>η</sup> δεύτερη σάρωση:
  - αν δυφίο χρήσης == 0
    - αν η σελίδα είναι λερωμένη, γράφεται στο δίσκο
      - σελίδες (τμημάτων) εισόδου/εξόδου
        - κλειδώνονται κατά την μεταφορά
  - το πλαίσιο αυτό προστίθεται στη λίστα των ελεύθερων
  - το πλαίσιο διατηρεί τα περιεχόμενά του, μέχρι να καταχωρηθεί σε άλλη διεργασία (γιατί?)

# Λογισμικό διαχείρισης μνήμης

## UNIX (4.2BSD)

### Εφαρμογή πολιτικής αντικατάστασης

- Κάθε 250 msec
  - Εξετάζεται ένας αριθμός πλαισίων (όχι όλα!)
    - Χρόνος δαίμονα  $< 10\%$  χρόνου της ΚΜΕ
  - συγκρίνεται το πλήθος των ελεύθερων πλαισίων ( $f$ ) με μια ορισμένη τιμή `lotsfree`
    - $lotsfree \cong 1/4$  πλαισίων του χάρτη που εξετάστηκαν
  - Αν  $f < lotsfree$ , αφυπνίζεται ο `pagedemon`
  - εφαρμόζει την πολιτική του (ελευθερώνει πλαίσια) μέχρι
    - $f == lotsfree$ , ή
    - $f ==$  πλήθος των πλαισίων που μπορεί ο χρονοπρογραμμαλ-τιστής της ΚΜΕ να έχει αποφασίσει ότι χρειάζονται (σε περίπτωση υπερφόρτωσης)
  - ο `pagedemon` αποκοιμίζεται
- μεγάλη αλληλεπίδραση μεταξύ
  - χρονοπρογραμματιστή μνήμης (μεσοχρόνιος) και
  - χρονοπρογραμματιστή ΚΜΕ (βραχυχρόνιος), βλ. Κεφ.9