

# Λειτουργικά Συστήματα (HY321)

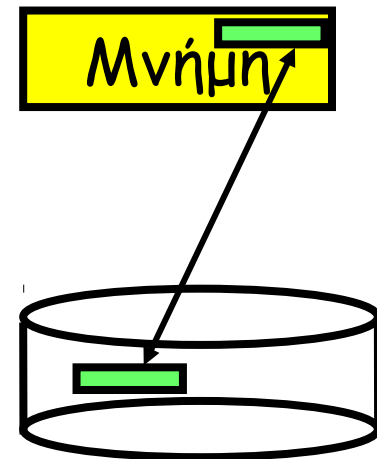
Διάλεξη 15:  
Caching Δίσκου, Αστοχίες,  
Συστήματα Αρχείων με  
Ημερολόγιο





# Η Χρήση Cache

- Τα γνωστά...
  - Αν το παρελθόν είναι παρόμοιο με το μέλλον...
  - ... μπορούμε να χρησιμοποιήσουμε cache
  - Ειδικά στους δίσκους, όπου κάθε προσπέλαση κοστίζει ακριβά !!!
- Η cache του δίσκου παρόμοια με την cache ιδεατών σελίδων (φυσική μνήμη)
- Τα αιώνια ερωτήματα:
  - Πόσο μεγάλη;
  - Τι να διώξω;
  - Τι γίνεται όταν γράφω;
  - Τι να προμεταφέρω;





# Μέγεθος Cache

- Πόσο μεγάλη;
  - Πόση φυσική μνήμη για cache δίσκου και πόση για cache σελίδων ιδεατής μνήμης;
    - Οι 2 κύριοι αντίπαλοι!
  - Τα παλιά τα χρόνια:
    - Προκαθορισμένα μεγέθη
    - Πρόβλημα;
      - Δεν προσαρμόζεται στις ανάγκες
    - Λύση;
  - Cache δυναμικά μεταβαλλόμενου μεγέθους
    - Πρόβλημα;
      - Συχνά έχουμε πολύ μεγάλα αρχεία...
      - Θα γεμίσουν την cache
      - Και θα προσπαθήσουν και να τη μεγαλώσουν
    - Λύση; Προκαθορισμένα μεγέθη cache
      - Ουφ, πάλι τα ίδια...
- Θυμάστε τι λέγαμε για τα πλαίσια μνήμης; (κοινά ή ανά χρήστη)
  - Ε, τα ίδια και εδώ...

# Πολιτική Απομάκρυνσης Από την Cache



- Γιατί μπορέσαμε να χρησιμοποιήσουμε cache;
  - Επειδή μέλλον ~ παρελθόν
  - Άρα LRU (κατά τα συνήθη)
    - Μάλιστα εδώ μπορούμε εύκολα να έχουμε την ιδανική LRU
    - Ναι αλλά...
      - Δεν είναι και τόσο καλή ιδέα πάντα
  - Αν κάποιο αρχείο είναι μεγαλύτερο από την cache;
    - LRU το χειρότερο που μπορείς να κάνεις
      - Γιατί;
    - !LRU (δηλαδή MRU) το καλύτερο που μπορείς να κάνεις
      - Περιορίζει τη «ζημιά»
    - Κάνε λοιπόν MRU για αυτά τα αρχεία (>cache)



# Αποθήκευση Αλλαγών

- Τι γίνεται όταν κάνω αλλαγή σε ένα αρχείο;
  - Με δεδομένο ότι το αρχείο είναι στην cache;
- **Απευθείας εγγραφή** και στο δίσκο (**write through**)
  - Όταν ένας ορμαθός στην cache μεταβληθεί, γράψτον και στο δίσκο
  - Μειονέκτημα;
    - Οι εγγραφές γίνονται αργέεεεες
  - Πλεονέκτημα;
    - Το ΣΑ είναι πάντα σε «ασφαλή» κατάσταση
  - Το κάνουν τα ΛΣ των PC, για πολλά αφαιρούμενα μέσα αποθήκευσης
    - Π.χ. για τη δισκέτα
    - Αλλά όχι για το USB stick (πώς το βλέπουμε;)
- **Καθυστερημένη εγγραφή** στο δίσκο (**write back**)
  - Όταν ένας ορμαθός στην cache μεταβληθεί, μάρκαρέ τον. Γράψε τον στο δίσκο αργότερα (με την ησυχία σου) ή όταν χρειαστεί να απομακρυνθεί από την cache
  - Πλεονεκτήματα;
    - Γρήγορες εγγραφές
    - Λιγότερες εγγραφές (αν γίνουν πάνω στο ίδιο σημείο οι νέες ακυρώνουν τις παλιές)
    - Ομαδοποιημένες εγγραφές (π.χ. πολλαπλές εγγραφές στον ίδιο ορμαθό ή γειτονικούς)
  - Μειονεκτήματα:
    - Όσο περισσότερο καθυστερείς τις εγγραφές, τόσο χειρότερο θα είναι το σκάσιμο (όταν συμβεί)

# Τα Προβλήματα του Write-Back



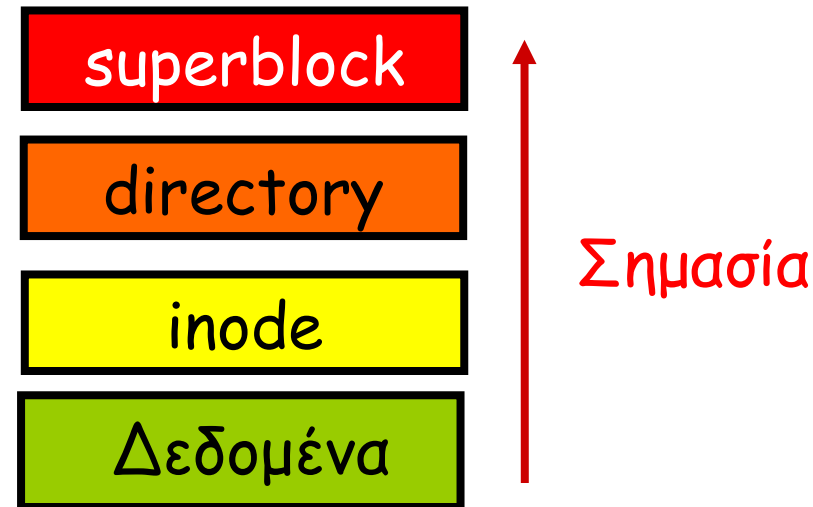
- Το πρόβλημα:
  - Αν έχουμε «σκάσιμο», όλες οι αλλαγές στην cache χάνονται
  - Όσο **περισσότερο αναβάλλεις** τις εγγραφές στο δίσκο, τόσο **ταχύτερος** είσαι, αλλά και τόσο μεγαλύτερη η πιθανή **ζημιά**
- Εγγραφές στο δίσκο σε 4 περιπτώσεις:
  - Όταν ένας ορμαθός απομακρύνεται από την cache (ότι κάνει και το σύστημα διαχείρισης ιδεατής μνήμης)
  - Όταν κλείνει ένα αρχείο
  - Όταν το ζητήσει ο χρήστης
  - Όταν περάσει κάποιος χρόνος (π.χ. 30 δευτερόλεπτα στο Unix)
- Η πολιτική write-back δε διατηρεί τη σειρά των εγγραφών στο δίσκο
  - Αστοχία = Το ΣΑ σε «περίεργη» κατάσταση
- Το ΛΣ πιθανόν να μην έχει επιλογή
  - Πολύ πιθανόν ο δίσκος να κάνει ούτως ή άλλως caching των εγγραφών



# Όλοι οι Ορμαθοί δε Γεννήθηκαν Ίσοι



- Κάποιοι είναι πιο κρίσιμοι από τους άλλους...
- Συνέπειες για το ΣΑ
  - Γράψε στο δίσκο τα αλλαγμένα μεταδεδομένα το συντομότερο δυνατό.
- Συνέπειες για τις εφαρμογές:
  - Για σημαντικά αρχεία, ζήτησε επισταμένα συγχρονισμό της cache με το δίσκο
  - Συμβαίνει συχνά στις βάσεις δεδομένων
    - Παρακάμπτουν εντελώς το ΣΑ





# Τι να Προμεταφέρω;

- Ιδανικά: Τους ορμαθούς που χρειαζόμαστε, και να είναι έτοιμοι ακριβώς όταν τους χρειαστούμε
  - Αν είχαμε αρκετό εύρος ζώνης προς το δίσκο και τη δυνατότητα να προβλέψουμε το μέλλον δε θα χρειαζόμασταν cache
    - Φέρε ότι χρειάζεσαι, φρόντισε να το έχεις τη στιγμή που θα το χρειαστείς και μετά πέταξέ το
- Παράδειγμα:
  - Εφαρμογή: Διαβάζει ορμαθό κάθε 1 msec, ξέρει τις μελλοντικές αναγνώσεις
  - Δίσκος: Μπορεί να δέχεται μια αίτηση κάθε 1 msec
  - Κάθε αίτηση παίρνει 10 msec
  - Πόσο μεγάλη cache χρειαζόμαστε;
    - Καθόλου. Στείλε κάθε ανάγνωση 10 msec πριν τη χρειαστείς
- Όμως... δεν ξέρουμε το μέλλον
  - Πληρώνουμε χώρο (cache) ως ποινή για την άγνοια





# «Χωρική» Τοπικότητα

- Αν προσπέρασα έναν αντικείμενο μάλλον θα προσπελάσω και τα **γειτονικά** του
  - π.χ., διάβασα έναν ορμαθό από αρχείο, μάλλον θα διαβάσω και τον επόμενο
  - Άρα αντί να φέρεις ένα ορμαθό, γιατί να μη φέρεις και τους  $n$  επόμενους;
- Κατ' ακρίβεια: Αν προσπέρασα έναν αντικείμενο μάλλον θα προσπελάσω και **τα σχετικά** με αυτό
  - Άρα βάλε τα σχετικά μαζί και ακολούθησε την παραπάνω τακτική
- Πόσο να είναι το " $n$ ";
  - Τι σημαίνει μεγάλο  $n$ ;
    - Κέρδος (χρόνου) αν είμαστε σωστοί
    - Χασούρα (χώρου) αν κάναμε λάθος πρόβλεψη
  - Συνήθως  $n < 64$  KB
  - Εναλλακτικά:
    - Φέρε μεγάλα κομμάτια (ούτως ή άλλως δεν κοστίζει ιδιαίτερα)
    - Πέταξε επιλεκτικά ότι δε φαίνεται να χρειάζεται

# Συστήματα Αρχείων με «Ημερολόγιο» (Logging)



- Θέλω να ξέρω τι συνέβαινε τη στιγμή της αστοχίας
  - Χρειάζομαι να αποκαταστήσω μόνο όσες δουλειές ήταν στη μέση
  - Συνήθως λίγες...
- Η μεγάλη ιδέα:
  - Ας κρατάμε ένα **ημερολόγιο (log)** των αιτήσεων προς το δίσκο.
  - Όταν έρθει η «κακιά η ώρα» κοιτάμε το ημερολόγιο
  - Βρίσκουμε τι έμεινε στη μέση, τι έχει δημιουργήσει πρόβλημα, τι χρειάζεται φτιάξιμο
- Ένα προβληματάκι καλέ κύριε...
  - Που θα το αποθηκεύσω;
  - Πρέπει να είναι σε μόνιμη (non-volatile) μνήμη
    - Δηλαδή στο δίσκο;



# Υλοποίηση

- Πρόσθεσε στο δίσκο μια περιοχή για το ημερολόγιο

Σύστημα Αρχείων	Ημ/γιο
-----------------	--------

- Γράψε πρώτα τις αλλαγές στο ημερολόγιο
  - Τεχνική γνωστή και ως *write-ahead logging* ή *journaling*
- Μετά γράψε τις αλλαγές και στο ΣΑ
- Οι αναγνώσεις απευθείας στο ΣΑ
- Επανάκαμψη από αστοχία:
  - Διάβασε το ημερολόγιο και διόρθωσε τυχόν μη νόμιμες καταστάσεις στο ΣΑ



# Και η Επίδοση;

- Δύο εγγραφές στο δίσκο (και μάλιστα σε διαφορετικά σημεία) για κάθε αλλαγή;
  - Άπαξ και γραφτεί στο ημερολόγιο, η αλλαγή δεν είναι ανάγκη να γραφτεί απευθείας και στο σύστημα αρχείων
    - Γιατί;
  - Είναι ασφαλές να χρησιμοποιήσουμε cache με πολιτική write-back
  - Συνήθης τρόπος λειτουργίας:
    - Η αλλαγές γίνονται στην cache (μνήμη) και καταγράφονται στο ημερολόγιο (δίσκος)
    - Ένωσε πολλαπλές αλλαγές στον ίδιο ορμαθό. Πολύ λιγότερες από 2 πραγματικές εγγραφές ανά αίτηση εγγραφής
- Σύγχρονες εγγραφές σε κάθε αλλαγή στο ΣΑ;
  - Οι εγγραφές του ημερολογίου γίνονται **ακολουθιακά** στο δίσκο, άρα ακόμα και οι σύγχρονες εγγραφές είναι **γρήγορες**
  - Για βέλτιστη επίδοση βάλτε το ημερολόγιο σε άλλο δίσκο

# Και... η Διαχείριση του Ημερολογίου;



- Πόσο μεγάλο θα είναι;
  - Όσο και το σύστημα αρχείων;
  - Μπορώ να επαναχρησιμοποιήσω το χώρο του;
- Μα το ημερολόγιο το χρειάζομαι **μόνο για να ανακάμψω** από αστοχίες
  - Να ξέρω τι ενέργειες είχαν γίνει αλλά δεν είχαν γραφτεί στο δίσκο
- Περιοδικός συγχρονισμός cache – δίσκου (**checkpointing**)
  - Μετά το συγχρονισμό μπορώ να σβήσω το μέχρι τώρα ημερολόγιο
  - Άρα χρειάζεται να είναι αρκετά μεγάλο για να κρατήσει μόνο τις αλλαγές που υπάρχουν στη μνήμη
- Τα περισσότερα ΣΑ με ημερολόγιο καταγράφουν στο ημερολόγιο **μόνο μεταδεδομένα** (περιγραφείς αρχείων, καταλόγους)
  - Όχι δεδομένα...
  - ... οπότε το μέγεθος του ημερολογίου γενικά μικρό



# Μορφή Ημερολογίου

- Και... τι ακριβώς καταγράφουμε;
- Υπάρχουν επιλογές...
  - Αντίγραφο του ορμαθού
    - π.χ ορμαθός καταλόγου, ορμαθός inode...
    - + εύκολο
    - - πολύς χώρος (και χρόνος)
    - Τι αντίγραφο; Από πότε;
      - Πριν την αλλαγή: Εύκολο να πάμε προς τα πίσω κατά την αποκατάσταση
      - Μετά την αλλαγή: Εύκολο να πάμε προς τα μπρος
      - Και τα 2: Εύκολο να πάμε προς οποιαδήποτε κατεύθυνση
  - Λογική αίτηση
    - π.χ. πρόσθεσε το όνομα “oslab” στον κατάλογο #222
    - + Λιγότερος χώρος
    - - Περισσότερη δουλειά στην αποκατάσταση
    - - Πιο δύσκολο / πονηρό



# Η Τάση στα ΣΑ με Ημερολόγιο

- Γρήγορη αποκατάσταση: Χρόνος  $O(\text{ενεργές πράξεις})$  αντί  $O(\text{μέγεθος δίσκου})$
- Καλύτερη επίδοση από τα απλά ΣΑ, με την προϋπόθεση της απαίτησης αξιοπιστίας
  - Αν πρέπει να γίνουν σύγχρονες εγγραφές, οι ακολουθιακές είναι πολύ γρηγορότερες από τις μη ακολουθιακές
  - Οι ασύγχρονες εγγραφές θα είναι γρηγορότερες από τις ακολουθιακές σύγχρονες, αλλά είναι επικίνδυνες...
- Παραδείγματα:
  - Windows NTFS.
  - Veritas.
  - Πολλοί ανταγωνιστές στο Linux
    - ext4fs, jfs, xfs, riserfs