

ΣΥΓΧΡΟΝΑ ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

3^η έκδοση

ANDREW S. TANENBAUM

Ασκήσεις -Ερωτήσεις
4^{ου} Κεφαλαίου

Άσκηση 4.1 (Πρ.3)

Είναι η κλήση συστήματος *open* για τα αρχεία εντελώς απαραίτητη στο UNIX; Ποιες θα ήταν οι συνέπειες αν η κλήση αυτή δεν υπήρχε;

Απάντηση 4.1 (Πρ.3)

- Η κλήση `open` αντιστοιχεί ένα μικρό ακέραιο (`file descriptor`) στο αρχείο που είναι ανοιχτό για επεξεργασία.
- Εάν δεν υπήρχε η κλήση `open`, θα ήταν αναγκαίο να καθοριστεί το όνομα του αρχείου που θα πρέπει να ανοιχθεί για να διαβαστεί *για κάθε κλήση της `read`*.
- Το σύστημα θα έπρεπε τότε να διαβάσει το `i-node` που αντιστοιχεί σε αυτό το αρχείο, παρόλο που θα μπορούσε να γίνει αυτό μόνο μία φορά.
- Ένα άλλο πρόβλημα είναι τότε θα γίνει πάλι η αντιγραφή του `i-node` στο δίσκο (τότε κλείνει το αρχείο).

Άσκηση 4.2 (Πρ.11)

Ένας τρόπος για να χρησιμοποιηθεί **συνεχής κατανομή** στο δίσκο χωρίς το πρόβλημα που προκαλείται από τα κενά είναι να συμπύσσεται ο δίσκος κάθε φορά που αφαιρείται ένα αρχείο. Από τη στιγμή που όλα τα αρχεία είναι συνεχή, η αντιγραφή ενός αρχείου απαιτεί μία αναζήτηση και μία καθυστέρηση λόγω περιστροφής κατά την ανάγνωση του αρχείου, οι οποίες ακολουθούνται από τη μεταφορά (που γίνεται με τη μέγιστη ταχύτητα). Η εγγραφή του αρχείου απαιτεί τις ίδιες εργασίες.

Αν υποθέσουμε ότι ο χρόνος αναζήτησης είναι **5 msec**, η καθυστέρηση λόγω περιστροφής **4 msec**, ο ρυθμός μεταφοράς **8 MB/sec**, και το μέσο μέγεθος αρχείου **8 KB**, πόσος χρόνος χρειάζεται για να διαβαστεί ένα αρχείο στην κύρια μνήμη και στη συνέχεια να γραφεί ξανά στο δίσκο σε νέα θέση;

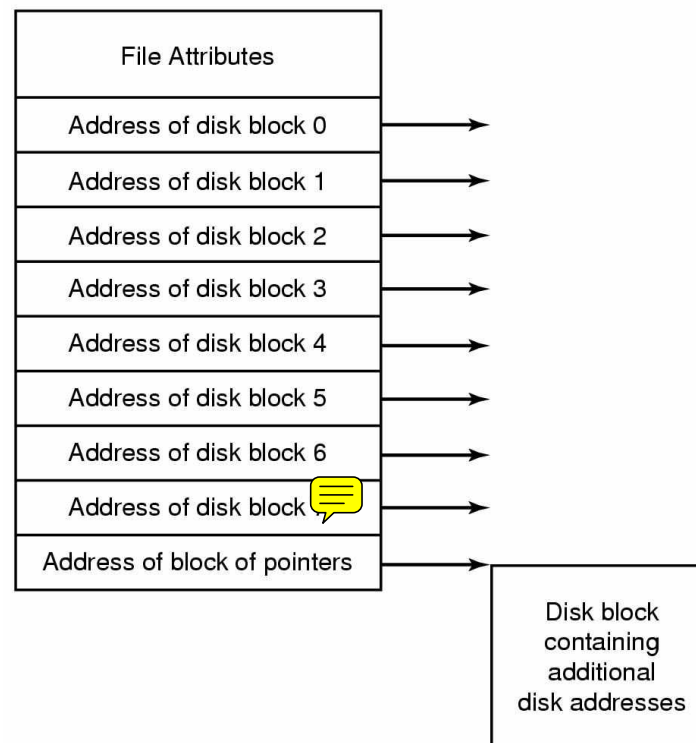
Με βάση τους ίδιους αριθμούς, πόσος χρόνος χρειάζεται για να συμπυκνωθεί ο μισός χώρος ενός δίσκου **16 GB**;

Απάντηση 4.2 (Πρ.11)

- Για να ξεκινήσει η ανάγνωση, χρειάζονται: $5+4 = 9$ msec.
- Για την ανάγνωση $8 \text{ KB} = 2^{13} \text{ bytes}$ με ταχύτητα μεταφοράς $8 \text{ MB/sec} = 2^{23} \text{ bytes/sec}$ χρειάζονται 2^{-10} sec (περ. 0,9765 msec).
- Συνολικός χρόνος ανάγνωσης 9,9765 msec.
- Συνολικός χρόνος ανάγνωσης και επανεγγραφής:
 $2 * 9,9765 = 19.953 \text{ msec}.$
- Η συμπίεση του μισού δίσκου συνολικού μεγέθους 16 GB, απαιτεί την εγγραφή 8 GB, (δηλ. με μέγεθος αρχείων 8 KB, σύνολο $2^{33} / 2^{13} = 2^{20}$ αρχεία).
- Χρόνος συμπίεσης $2^{20} * 19.953 \text{ msec}$ περίπου 5,8 ώρες.

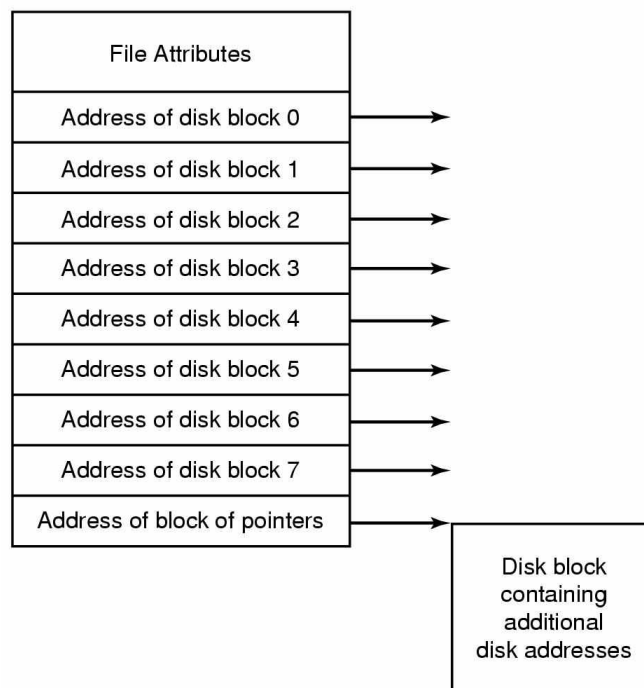
Άσκηση 4.3 (Πρ.15)

Θεωρήστε τον κόμβο i της Εικόνας 4-13. Αν υποθέσουμε ότι περιέχει 10 άμεσες διευθύνσεις των 4 byte, ενώ όλα τα μπλοκ καταλαμβάνουν 1024 byte, ποιο είναι το μεγαλύτερο αρχείο που μπορεί να υπάρξει;



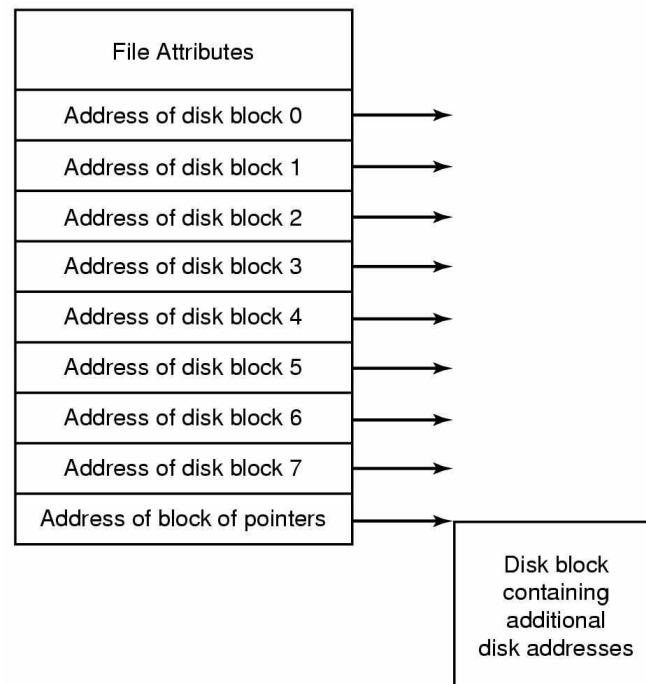
Απάντηση 4.3 (Πρ.15)

- Κάθε έμμεσο μπλοκ μπορεί να χωρέσει $1024/4=256$ διευθύνσεις δίσκου.
- Μαζί με τις $10 - 1 = 9$ άμεσες διευθύνσεις δίσκου, συνολικά 265 μπλοκ.
- Εφόσον κάθε μπλοκ είναι 1 KB, το μέγιστο μέγεθος αρχείου είναι 265 KB.



Άσκηση 4.4 (Πρ.16)

Έχει προταθεί ότι η απόδοση μπορεί να βελτιωθεί και να εξοικονομηθεί χώρος στο δίσκο αν τα δεδομένα των μικρών αρχείων αποθηκεύονται μέσα στον κόμβο i . Όσο αφορά τον κόμβο i της Εικόνας 4-13, πόσα byte δεδομένων μπορούν να αποθηκευθούν μέσα σε αυτόν;



Απάντηση 4.4 (Πρ.16)

- Θα πρέπει να χάσουμε κάποιο χώρο μέσα στον i-node, ώστε να σηματοδοτηθεί ότι οι δείκτες του i-node περιλαμβάνουν στην πραγματικότητα δεδομένα, και όχι δείκτες.
 - Εάν περισσεύουν κάποια bit μέσα στο πεδίο attributes, θα μπορούσαν να χρησιμοποιηθούν για να δείξει αν ο i-node περιλαμβάνει δείκτες ή δεδομένα. Σε αυτή την περίπτωση και οι 9 δείκτες θα χρησιμοποιηθούν για αποθήκευση δεδομένων. Για **k bytes ανά δείκτη**, το αρχείο που θα αποθηκευτεί εκεί μπορεί να είναι μέχρι **$9*k$ bytes**.
 - Εάν δεν μπορέσουμε να χρησιμοποιήσουμε κάποιο bit από τα χαρακτηριστικά, ο 1^{ος} δείκτης θα χαθεί για αυτό το σκοπό, οπότε το μέγιστο μέγεθος περιορίζεται σε **$8*k$ bytes**.

Άσκηση 4.5 (Πρ.20)

Μετά από την πρώτη διαμόρφωση ενός διαμερίσματος δίσκου, η αρχή ενός **χάρτη bit** ελεύθερου χώρου είναι η εξής: 1000 0000 0000 0000 (το πρώτο μπλοκ χρησιμοποιείται για το βασικό κατάλογο). Το σύστημα ψάχνει πάντα για ελεύθερα μπλοκ ξεκινώντας από το μπλοκ με το μικρότερο αριθμό. Επομένως, μετά από την εγγραφή του αρχείου *A* το οποίο χρησιμοποιεί 6 μπλοκ, ο χάρτης bit γίνεται ως εξής: 1111 1110 0000 0000. Δώστε τη μορφή του χάρτη bit μετά από τις επόμενες ενέργειες:

- (α) Γράφεται το αρχείο *B*, το οποίο χρησιμοποιεί 5 μπλοκ.
- (β) Διαγράφεται το αρχείο *A*.
- (γ) Γράφεται το αρχείο *Γ*, το οποίο χρησιμοποιεί 8 μπλοκ,
- (δ) Διαγράφεται το αρχείο *B*.

Απάντηση 4.5 (Πρ.20)

Αρχική κατάσταση: 1111 1110 0000 0000

(α) Εγγραφή του B (5 μπλοκ) : 1111 1111 1111 0000

(β) Διαγραφή του A (6 μπλοκ) : 1000 0001 1111 0000

(γ) Εγγραφή του Γ (8 μπλοκ) : 1111 1111 1111 1100

(δ) Διαγραφή του B (5 μπλοκ) : 1111 1110 0000 1100

Άσκηση 4.6 (Πρ.25)

Έχει προταθεί το πρώτο τμήμα κάθε αρχείου στο UNIX να διατηρείται στο ίδιο μπλοκ δίσκου με τον αντίστοιχο κόμβο-*i*. Τι καλό θα προκύψει από αυτό;

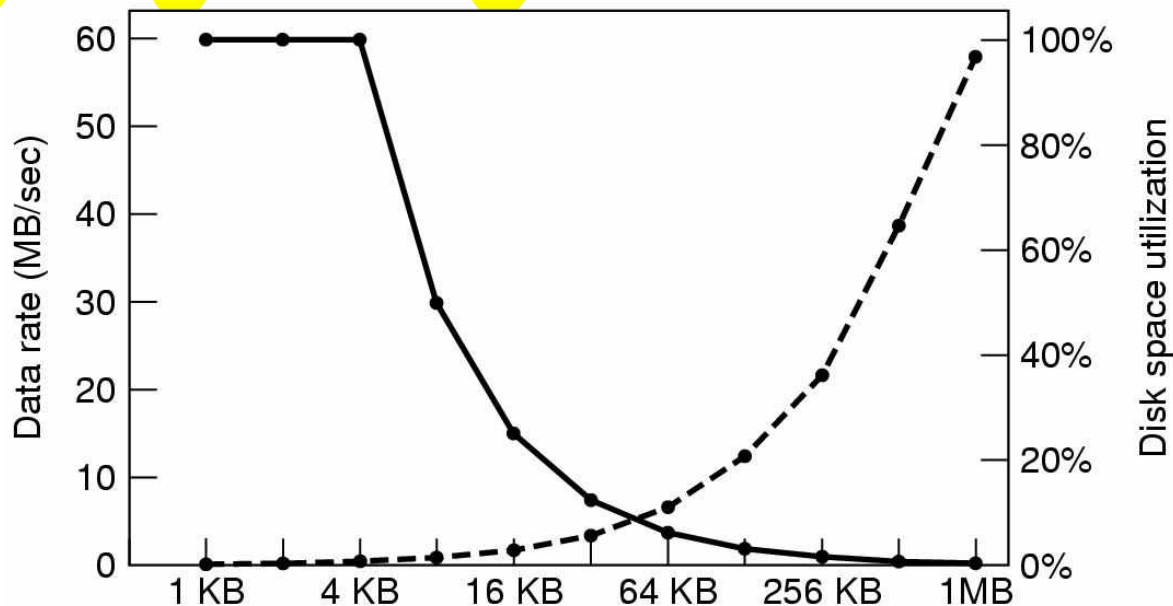
Απάντηση 4.6 (Πρ.25)

- Παρούσα κατάσταση: Για αρχείο με k μπλοκ, χρειάζονται $k+1$ προσβάσεις στο δίσκο (ή 1 για την ανάγνωση του i-node)
- Εάν ο i-node βρισκόταν στο ίδιο μπλοκ δίσκου με το 1^ο τμήμα του αρχείου, θα υπήρχε όφελος μίας πρόσβασης στο δίσκο.
- Εάν το αρχείο είναι μικρό και χωράει στο ίδιο μπλοκ με τον i-node, θα χρειάζεται μόνο μία πρόσβαση στο δίσκο!

Άσκηση 4.7 (Πρ.28)

Θεωρήστε την ιδέα πίσω από την Εικόνα 4-21, αλλά τώρα για ένα δίσκο με μέσο χρόνο αναζήτησης **8 msec**, ρυθμό περιστροφής **15.000 rpm**, και **262.144 byte ανά τροχιά**.

Ποιοί είναι οι ρυθμοί μεταφοράς δεδομένων για μεγέθη μπλοκ **1 KB**, **2 KB**, και **4 KB** αντίστοιχα;



Απάντηση 4.7 (Πρ.28)

- Ρυθμός περιστροφής 15.000 rpm: σε 1min γίνονται 15.000 στροφές, άρα η μία περιστροφή σε:

$$(60 * 10^3) / 15.000 \text{ msec} = 4 \text{ msec ανά περιστροφή}$$

- Ο μέσος χρόνος προσπέλασης σε msec για την ανάγνωση μπλοκ μεγέθους k byte, θα είναι:

Χρ. αναζήτησης + χρ. περιστροφής + χρ. μεταφοράς μπλοκ

$$8 + 2 + (k / 262144) \times 4 \text{ msec.}$$

- Για μπλοκ 1 KB, 2 KB, and 4 KB, ο χρόνος προσπέλασης είναι αντίστοιχα 10,015625 msec, 10,03125 msec, και 10,0625 msec.
- Συνεπώς, ο ρυθμός μεταφοράς δεδομένων είναι περίπου 99,850 KB/sec, 199,377 KB/sec, και 397,515 KB/sec, αντίστοιχα.

Άσκηση 4.8 (Πρ.29)

Ένα συγκεκριμένο σύστημα αρχείων έχει μπλοκ δίσκου με μέγεθος 2 KB. Η κεντρική τιμή του μεγέθους αρχείου είναι 1 KB.

Αν όλα τα αρχεία ήταν ακριβώς 1 KB, τι ποσοστό χώρου του δίσκου θα πήγαινε χαμένο;

Πιστεύετε ότι οι απώλειες σε ένα πραγματικό σύστημα αρχείων είναι μεγαλύτερες ή μικρότερες από το σύστημα αυτό;

Εξηγήστε την απάντησή σας.

Απάντηση 4.8 (Πρ.29)

- Για αρχεία του 1 KB, τότε για κάθε μπλοκ των 2 KB, χάνεται χώρος 1 KB (απώλεια χώρου 50%).
- Στην πράξη κάθε σύστημα αρχείων έχει πολλά μικρά αρχεία και **μεγάλα αρχεία** τα οποία χρησιμοποιούν το χώρο του δίσκου πιο αποδοτικά.
- Παράδειγμα:
 - έστω αρχείο με μέγεθος $(1 \text{ MB} + 1 \text{ byte}) = 2^{20} + 1 = 1.048.577 \text{ byte}$.
 - θα χρησιμοποιούσε $(2^{20} / 2 * 2^{10}) + 1 = 2^9 + 1 = 513 \text{ μπλοκ}$
 - Άρα ποσοστό αξιοποίησης δίσκου:
 $1.048.577 / (513 * (2 * 2^{10})) = 1.048.577 / 1.050.624 = 0,99805$

Άσκηση 4.9 (Πρ.31)

Τα αρχεία στο MS – DOS πρέπει να ανταγωνιστούν για την απόκτηση χώρου στον πίνακα FAT-16 ο οποίος βρίσκεται στη μνήμη. Αν κάποιο αρχείο χρησιμοποιεί k καταχωρίσεις, τότε οι συγκεκριμένες k καταχωρίσεις δεν είναι διαθέσιμες σε οποιοδήποτε άλλο αρχείο, τι περιορισμούς θέτει αυτό στο συνολικό μήκος όλων μαζί των αρχείων;

Απάντηση 4.9 (Πρ.31)

- Περιορίζει το άθροισμα του μεγέθους όλων των αρχείων να είναι μικρότερο από το δίσκο.
- Δεν είναι σημαντικός περιορισμός. Εάν τα αρχεία είναι αθροιστικά μεγαλύτερα από το δίσκο, δεν υπάρχει έτσι κι'αλλιώς χώρος αποθήκευσης.

Άσκηση 4.10 (Πρ. 33)

Πόσες λειτουργίες δίσκου χρειάζονται για να μεταφερθεί ο κόμβος-ι του αρχείου **/usr/ast/courses/os/handout.t** ;

Υποθέστε ότι ο κόμβος-ι που αντιστοιχεί στον βασικό κατάλογο βρίσκεται στη μνήμη και ότι τίποτα άλλο σχετικά με την παραπάνω διαδρομή δεν βρίσκεται στη μνήμη.

Υποθέστε επίσης ότι όλοι οι κατάλογοι βρίσκονται σε ένα μπλοκ δίσκου.

Απάντηση 4.10 (Πρ.)

- Χρειάζονται οι ακόλουθες προσβάσεις ανάγνωσης στο δίσκο (συνολικά 10).
- Κατάλογος του /
- i-node του */usr*
- Κατάλογος του */usr*
- i-node του */usr/ast*
- Κατάλογος του */usr/ast*
- i-node του */usr/ast/courses*
- Κατάλογος του */usr/ast/courses*
- i-node του */usr/ast/courses/os*
- Κατάλογος του */usr/ast/courses/os*
- i-node του */usr/ast/courses/os/handout.t*