ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΡΧΕΙΩΝ

1η άσκηση

Ημερομηνία παράδοσης: 27 Μαρτίου 2017
Η άσκηση είναι ατομική

Επεξεργασία Αρχείων

Σκοπός της άσκησης είναι η εξοικείωση με την απόδοση μεθόδων αναζήτησης στον δίσκο. Η πληροφορία οργανώνεται σε σελίδες δίσκου. Για τους σκοπούς της άσκησης το μέγεθος της σελίδας είναι 512 bytes. Υποθέτουμε ότι η σελίδα αποθηκεύει σε δυαδική μορφή (και όχι text) ένα πεδίο 128 ακεραίων αριθμών (4 bytes ο καθένας). Η επεξεργασία του αρχείου δεν γίνεται απευθείας στον δίσκο αλλά μεταφέροντας στην κεντρική μνήμη κάθε φορά μία σελίδα. Η σελίδα διαβάζεται σε έναν buffer μεγέθους 512 bytes που μετατρέπεται σε πεδίο 128 ακεραίων. Η επεξεργασία γίνεται στο πεδίο στην κεντρική μνήμη και αν χρειαστεί, το πεδίο μετατρέπεται σε buffer που ξαναγράφεται ως σελίδα δίσκου πίσω στο αρχείο. Η απόδοση μιας μεθόδου επεξεργασίας εξαρτάται από τον αριθμό προσβάσεων σε σελίδες δίσκου (disk accesses).

Ένα αρχείο μπορούμε να το διαβάσουμε σειριακά, διαβάζοντας κάθε φορά μία σελίδα (την μία σελίδα μετά την άλλη) και μεταφέροντας την κάθε σελίδα στην κεντρική μνήμη. Μπορούμε να αλλαξουμε την σειρά ανάγνωσης των σελίδων αν ξέρουμε πως μπορούμε να φορτώσουμε στην μνήμη την σελίδα που είναι σε κάποια θέση του αρχείου χωρίς προηγουμένως να πρέπει να διαβασουμε τις προηγούμενες. Αυτό γίνεται με μια εντολή seek για μετάβαση στην θέση της σελίδας που μας ενδιαφέρει και μία εντολή read για το διάβασμά της από την θέση μετάβασης. Η εκτέλεση της εντολής read προωθεί τον δείκτη αρχείου στην επόμενη σελίδα.

Εισαγωγή στοιχείων

Δημιουργήστε ένα **ταξινομημένο** αρχείο με $\mathbf{N = 10^7}$ ακεραίους αριθμούς (κλειδιά) στο δίσκο με τιμές από 1 έως 10^7. Στην διάρκεια της εισαγωγής, σε κάθε σελίδα φορτώνουμε 128 κλειδιά σε ένα (buffer) μεγέθους σελίδας. Γράψτε τον buffer στον δίσκο. Φορτώνουμε τον buffer με τα επόμενα κλειδιά μέχρι να γραφτούν όλα τα N κλειδιά στον δίσκο.

Α. Σειριακή αναζήτηση στο αρχείο για τυχαίο κλειδί (2 μονάδες)

Κάθε ερώτηση εκφράζεται με μία τυχαία τιμή κλειδιού στο διάστημα 1 έως 10^7. Το κλειδί παράγεται από μια γεννήτρια τυχαίων αριθμών που παράγει αριθμούς στο διάστημα τιμών από 1 έως N. Ψάξτε το αρχείο ως εξής: Ανοίξτε το αρχείο και διαβάστε την πρώτη σελίδα του αρχείου στον buffer της κεντρικής μνήμης (αυτο κοστίζει μία πρόσβαση στο δίσκο). Ψάξτε στον buffer με δυαδική αναζήτηση (binary search) στην κεντρική μνήμη. Αν το κλειδί που ψάχνετε δεν υπάρχει στον buffer τότε επαναλαμβάνεται ή ίδια διαδικασία για την επόμενη σελίδα του δίσκου, και αν χρειαστεί ξανά μέχρι να εξαντληθεί το αρχείο. Μετρήστε τον αριθμό προσβάσεων που χρειάστηκαν μέχρι να βρεθεί το κλειδί. Επαναλάβετε το ίδιο για 10.000 αναζητήσεις τυχαίων κλειδιών. Μετρήστε τον μέσο αριθμό προσβάσεων για όλες τις αναζητήσεις.

Β. Δυαδική αναζήτηση στο αρχείο για τυχαίο κλειδί (2 μονάδες)

Όπως στο προηγούμενο ερώτημα θα κάνετε 10.000 ερωτήσεις με τυχαίες τιμές κλειδιών και θα μετρήστε τον μέσο αριθμό προσβάσεων. Για κάθε ερώτηση, εκμεταλλευτείτε ότι το αρχείο είναι ταξινομημένο: διαβάστε αρχικά την μεσαία σελίδα του αρχείου στην κεντρική μνήμη. Εξετάστε αν το κλειδί είναι στην σελίδα (με δυαδική αναζήτηση). Αν υπάρχει, η αναζήτηση σταματάει. Αν το κλειδί που ψάχνετε είναι μικρότερο από τον μικρότερο αριθμό της σελίδας, τότε φέρτε στην κεντρική μνήμη την μεσαία σελίδα του αριστερού μισού του αρχείου. Αν το κλειδί είναι μεγαλύτερο από τον μεγαλύτερο αριθμό της σελίδας, φέρτε στην κεντρική μνήμη την μεσαία σελίδα του δεξιού μισού του αρχείου. Συνεχίστε με τον ίδιο τρόπο την αναζήτηση.

Γ. Δυαδική Αναζήτηση με ομαδοποίηση των ερωτήσεων (2 μονάδες)

Ο μηχανισμός αναζήτησης βασίζεται σε Δυαδική Αναζήτηση του ερωτήματος Β. Το σύστημα συγκεντρώνει όλες τις ερωτήσεις και τις επεξεργάζεται όλες μαζί αφού τις ταξινομήσει σε αύξουσα σειρά κλειδιών. Αν η σελίδα του κλειδιού μιας ερώτησης βρίσκεται ήδη στην κεντρική μνήμη τότε το σύστημα δεν χρειάζεται να την ξαναφέρει. Κάτω από προϋποθέσεις, αυτή η στρατηγική απαιτεί λιγότερες προσβάσεις στο αρχείο (κατά μέσο όρο) από την περίπτωση που κάθε ερώτηση θα εξυπηρετηθεί ανεξάρτητα από την επόμενη. Μετρήστε τον μέσο αριθμό προσβάσεων για 10.000 αναζητήσεις τυχαίων κλειδιών.

Δ. Δυαδική Αναζήτηση με χρήση προσωρινής μνήμης (2 μονάδες)

Ο μηχανισμός αναζήτησης βασίζεται σε Δυαδική Αναζήτηση του ερωτήματος Β. Το σύστημα διατηρεί στην κεντρική μνήμη μία ουρά (queue) με Κ (Κ=1, 10, 100, 1.000) από τις σελίδες που έχει φέρει από το αρχείο, με την σειρά που τις έχεις φέρει. Δηλαδή αν μετά από διάστημα χρήσης ξεπεραστεί ο αριθμός Κ, τότε σβήνεται η πιο παλιά σελίδα.
Μια άλλη στρατηγική θα ήταν να σβήνεται η σελίδα που χρησιμοποιήθηκε λιγότερο στην διάρκεια της χρήσης του συστήματος (μην το υλοποιήσετε, σκεφτείτε όμως πως θα μπορούσατε να το υλοποιήσετε). Εξηγήστε με σαφήνεια στο ερώτημα Ε (παρακάτω) πως η επιλογή της μιας ή της άλλης στρατηγικής επηρεάζει την απόδοση της μεθόδου (αν την βελτιώνει ή όχι και για ποιο λόγο).
Μετρήστε τον μέσο αριθμό προσβάσεων για 10.000 αναζητήσεις τυχαίων κλειδιών για Κ =1, 50 και 100 (δηλαδή θα δώσετε 3 μετρήσεις).

Ε. Τεκμηρίωση των Αποτελεσμάτων (2 μονάδες)

Συγκεντρώστε τα αποτελέσματα στον παρακάτω και προσπαθήστε να δικαιολογήστε την απόδοση κάθε μεθόδου.

Μέθοδος	**Α**	**Β**	**Γ**	**Δ (Κ=1)**	**Δ (Κ=50)**	**Δ (Κ=100)**
Απόδοση						

Παραδοτέα: Ένα συμπιεσμένο zip αρχείο που περιέχει ότι ζητείται παρακάτω:

- Ο κώδικας περιέχει συνοπτικά σχόλια που εξηγούν την υλοποίηση.
- Μία έκθεση που περιγράφει σε 1-2 σελίδες πως φτιάχτηκε ο κώδικας (δηλ. για κάθε ερώτημα ποια είναι η γενική ιδέα της λύσης σε 3-4 προτάσεις), υπάρχουν σαφείς οδηγίες μετάφρασης από compiler και εκτέλεσης, τι λάθη έχει (αν έχει, περιπτώσεις που δεν

δουλεύει το πρόγραμμα, ή περιπτώσεις που κάνει περισσότερα από όσα σας ζητεί η άσκηση, τι χρησιμοποιήσατε από έτοιμα προγράμματα ή πηγές πληροφόρησης. Υποδείξετε ακόμα και πηγές στο WWW όπως Wikipedia ή ακόμα και συναδέλφους που σας βοήθησαν στην άσκηση.

- Για το ερώτημα Ε πρέπει να υπάρχει τεκμηρίωση των αποτελεσμάτων με σαφήνεια.
- Εκτός των παραπάνω, οι ασκήσεις βαθμολογούνται με άριστα εφόσον:
    - Το zip είναι πλήρες
    - Οι κώδικες περνούν από compiler και εκτελούνται κανονικά και σωστά σε windows ή Linux περιβάλλον
    - Ο κώδικάς σας δουλεύει για οποιοδήποτε ταξινομημένο αρχείο αριθμών που θα δοθεί ως είσοδος.