

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΡΧΕΙΩΝ

1^η άσκηση

Ημερομηνία παράδοσης: 26 Μαρτίου 2018

Η άσκηση είναι ατομική

Επεξεργασία Αρχείων

Σκοπός της άσκησης είναι η εξοικείωση με χρήση αρχείων δίσκου. Η πληροφορία οργανώνεται σε σελίδες δίσκου. Για τους σκοπούς της άσκησης το μέγεθος της σελίδας είναι $page_size=512$ 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. Κάθε σελίδα έχει έναν αριθμό θέσης pos (αριθμός χαρακτήρων από την αρχή του αρχείου $\times page_size$). Η πρόσβαση στην σελίδα γίνεται με μια εντολή seek (για μετάβαση στην θέση της σελίδας που 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 αρχείο με $N = 10^7$ ακεραίους τυχαίους αριθμούς (κλειδιά) στο δίσκο με τιμές από 1 έως 10^7 . Οι τιμές παράγονται από μία γεννήτρια τυχαίων αριθμών. Στην διάρκεια της εισαγωγής, σε κάθε σελίδα φορτώνουμε 128 κλειδιά σε ένα (buffer) μεγέθους σελίδας. Γράφουμε τον buffer στον δίσκο όταν γεμίσει. Φορτώνουμε τον buffer με τα επόμενα κλειδιά μέχρι να γραφτούν όλα τα N κλειδιά στον δίσκο.

A. Κλάση Αρχείο (2 μονάδες)

Κατασκευάστε την κλάση «**FileManager**» που με λειτουργίες:

- **FileHandle:** αναπαριστάνει ένα ανοικτό αρχείο, την τρέχουσα θέση δείκτη αρχείου, αριθμό σελίδων στο αρχείο, είδος πληροφορίας που αποθηκεύεται και οποιαδήποτε άλλη χρήσιμη πληροφορία που θα χρειαστεί μια εφαρμογή που θα χρησιμοποιήσει το αρχείο. Αυτή η πληροφορία μπορεί γράφεται στην πρώτη σελίδα του αρχείου.
- **CreateFile:** αρχικοποίηση αρχείου, δηλαδή δημιουργεί ένα αρχείο χωρίς δεδομένα με μια σελίδα στην αρχή που αποθηκεύει την πληροφορία που υπάρχει στο FileHandle. Το όνομα του αρχείου στο δίσκο δίνεται ως όρισμα. Ο αριθμός σελίδων παίρνει τιμή 0.

- **OpenFile:** Άνοιγμα αρχείου για ανάγνωση ή εγγραφή, επιστρέφει τον αριθμό σελίδων του αρχείου. Το όνομα του αρχείου στο δίσκο δίνεται ως όρισμα.
- **ReadBlock:** Ανάγνωση σελίδας στο buffer (στην κεντρική μνήμη) στη θέση αρχείου που δίνεται ως παράμετρος.
- **ReadNextBlock:** Ανάγνωση επόμενης σελίδας (ως προς την τρέχουσα θέση) στο buffer
- **ReadPrevBlock:** Ανάγνωση προηγούμενης σελίδας (ως προς την τρέχουσα θέση) στο buffer.
- **WriteBlock:** Εγγραφή του buffer στην θέση αρχείου που δίνεται ως παράμετρος. Δεν αυξάνεται ο αριθμός σελίδων του αρχείου αν η εγγραφή γίνεται πάνω σε σελίδα που υπάρχει ήδη.
- **WriteNextBlock:** Εγγραφή buffer στην επόμενη σελίδα αρχείου (ως προς την τρέχουσα θέση). Δεν αυξάνεται ο αριθμός σελίδων του αρχείου αν η εγγραφή γίνεται πάνω σε σελίδα που υπάρχει ήδη.
- **AppendBlock:** Εγγραφή buffer στο τέλος του αρχείου. Αυξάνεται ο αριθμός σελίδων του αρχείου.
- **DeleteBlock:** Διαγραφή μίας σελίδας από το αρχείο. Η διαγραφή γίνεται αντιγράφοντας την τελευταία σελίδα του αρχείου στην θέση της σελίδας που διαγράφεται και μειώνοντας κατά ένα τον αριθμό σελίδων του αρχείου. Πρέπει να ενημερώνεται και η FileHandle για τον νέο αριθμό σελίδων.
- **CloseFile:** Αντιγραφή του αρχείου σε ένα νέο με τα δεδομένα που έχει εκείνη την στιγμή το FileHandle. Το όνομα του αρχείου στο δίσκο δίνεται ως όρισμα.

Κάθε λειτουργία επιστρέφει 0 αν για κάποιο λόγο αποτύχει και 1 αν επιτύχει.

B. Αναζήτηση στο Αρχείο (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 (αυτό κοστίζει μία πρόσβαση στο δίσκο). Ψάξτε στον buffer στην κεντρική μνήμη. Αν το κλειδί 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. Αν το κλειδί επαναλαμβάνεται και σε άλλη σελίδα θα βρείτε μόνο την πρώτη. Επαναλάβετε το ίδιο για 10.000 αναζητήσεις τυχαίων κλειδιών. Μετρήστε τον μέσο αριθμό προσβάσεων για όλες τις αναζητήσεις.

Γ. Κλάση Buffer Pool (2 μονάδες)

Είναι μία κλάση που παριστάνει μία λίστα σελίδων (τύπου buffer). Πριν εκτελεστεί οποιαδήποτε ανάγνωση σελίδας από τον δίσκο ελέγχουμε αν υπάρχει ήδη στο buffer pool. Αν υπάρχει τότε δεν γίνεται ανάγνωση από το αρχείο. Η αναζήτηση γίνεται δίνοντας ως όρισμα την θέση της σελίδας στο αρχείο. Με αυτό το τρόπο κερδίζουμε χρόνο μειώνοντας τις αναζητήσεις στον δίσκο. Η κλάση Buffer Pool μπορεί να έχει μέχρι K σελίδες (όπου το K είναι παράμετρος).

Κατασκευάστε την κλάση «**PoolManager**» που με λειτουργίες:

- **CreatePool:** Αρχικοποίηση της λίστας σελίδων του αρχείου που δίνεται ως όρισμα και της πληροφορίας που αφορούν το αρχείο (διαβάζοντας την πρώτη σελίδα του αρχείου).

- **SearchPool:** Αναζήτηση στην λίστα (με βάση την θέση της σελίδας στο αρχείο).
- **InsertPool:** Εισαγωγή νέας εγγραφή σελίδας αρχείου στην λίστα (αν δεν υπάρχει ήδη). Αν δεν υπάρχει η σελίδα η εισαγωγή γίνεται στην αρχή της λίστας. Επομένως οι παλαιότερες σελίδες βρίσκονται στο τέλος της λίστας. Αν η λίστα έχει ήδη K σελίδες, τότε διαγράφεται η παλαιότερη (αυτή στο τέλος της λίστας).
- **DeletePool:** Διαγραφή σελίδας από την λίστα (αν υπάρχει)
- **FreePool:** Διαγραφή της λίστας σελίδων.

Δ. Αναζήτηση με χρήση Buffer Pool (2 μονάδες)

Όπως στο ερώτημα B θα κάνετε 10.000 ερωτήσεις με τυχαίες τιμές κλειδίων και θα μετρήσετε τον μέσο αριθμό προσβάσεων στο δίσκο. Για κάθε ερώτηση, εκμεταλλευτείτε το Buffer Pool για να μειώσετε τον αριθμό προσβάσεων στο δίσκο. Να επαναλάβετε τις ίδιες μετρήσεις με τιμή K = 1.000 και 10.000.

Ε. Τεκμηρίωση των Αποτελεσμάτων (2 μονάδες)

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

Μέθοδος	B	Δ (K=1.000)	Δ (K=10.000)
Απόδοση			

Παραδοτέα: Ένα συμπίεμένο 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. Υποδείξτε ακόμα και πηγές στο WWW όπως Wikipedia ή ακόμα και συναδέλφους που σας βοήθησαν στην άσκηση.
- Για το ερώτημα E 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.