***Υλοποίηση Συστημάτων Βάσεων Δεδομένων***

***Δεύτερη Εργασία***

***Γιάννης Δαλιάνης***,***1115201700027***

***Μιχάλης Τατάς,***

***Μιχάλης Φωτιάδης,***

Το πρώτο block κάθε αρχείου κατακερματισμού περιλαμβάνει ως ειδική πληροφορία το string “**HT**”, το οποίο δηλώνει την ταυτότητα του αρχείου. Το αρχείο αρχικοποιείται με την **HT\_CreateIndex**, όπου δεσμεύεται τόσο το πρώτο block, δηλαδή το block πληροφορίας όσο και τα block ευρετηρίου. Η δομή του block πληροφορίας είναι η εξής:

|  |  |  |  |
| --- | --- | --- | --- |
| ΗΤ | Πλήθος εγγραφών αρχείου | Πλήθος κουβάδων αρχείου | Κενός χώρος |

Τα block που δεσμεύονται αμέσως μετά είναι τα block ευρετηρίου. Η δομή ενός block ευρετηρίου είναι:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Αριθμός επόμενου block ευρετηρίου | 2 | 3 | 4 | . . . | . . . |

Ουσιαστικά από το δύο και μετά κάθε αριθμός αντιστοιχεί σε ένα bucket. Η αρίθμηση γίνεται στο διάστημα [2, buckets+2]. Θεωρούμε ότι το 1 αντιστοιχεί στο block πληροφορίας. Για κάθε bucket αρχικοποιώ και ένα block, το οποίο θα είναι το πρώτο block εγγραφών για τη συγκεκριμένη τιμή hash.

Η συνάρτηση **hashFunctions** δέχεται ως παραμέτρους τον αριθμό του αρχείου και το id μιας εγγραφής. Παίρνει από το πρώτο μπλοκ του αρχείου τον αριθμό των buckets και επιστρέφει την hashing τιμή που προκύπτει από το id σύμφωνα με τη συνήθη πράξη **(id mod buckets).**

Στη συνάρτηση **HT\_InsertEntry** δίνεται ένα record. Σε τμήμα κώδικα που έχει σχολιαστεί, γίνεται ένας έλεγχος για την περίπτωση που το ευρετήριο χρειάζεται να αλλαχτεί με rehash. Η rehash δεν έχει υλοποιηθεί για την εργασία. Λόγω της εισαγωγής record, o αριθμός των records στο block πληροφορίας αυξάνεται κατά ένα. Από το ευρετήριο και χρησιμοποιώντας τη hashFunctions παίρνω τον αριθμό του block στο οποίο πρέπει να εισαχθεί η νέα εγγραφή. Σε κάθε block εγγραφής χωράνε μέχρι 8 εγγραφές. Δομή μπλοκ εγγραφής:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Εγγραφές block | Επόμενο block | Εγγραφή 1 | Εγγραφή 2 | . . . | Εγγραφή 8 |

Όταν κάνω εισαγωγή εγγραφής, διατρέχω την αλυσίδα των block για τη συγκεκριμένη αξία hash, μέχρι να βρω ένα block το οποίο στη θέση επόμενου block έχει το 0, κάτι που σημαίνει ότι είναι το τελευταίο block της αλυσίδας. Σε περίπτωση που χωράει στο block η εγγραφή, γράφεται μετά την τελευταία υπάρχουσα και αυξάνεται κατά 1 το πλήθος εγγραφών του block. Σε διαφορετική περίπτωση, δεσμεύεται ένα νέο block και στο αμέσως προηγούμενο block τοποθετώ τον αριθμό του καινούριου block που δεσμεύτηκε ως επόμενο block. Βάζω στο καινούριο block ως πλήθος εγγραφών το 1 και εισάγω την εγγραφή.

Η συνάρτηση **HT\_PrintAllEntries** καλεί για κάθε bucket του ευρετηρίου τη συνάρτηση **HT\_PrintBlockChain,** η οποία δέχεται τον αριθμό id του αρχείου, τον αριθμό του block εγγραφών και το id της εγγραφής που θέλω να εκτυπώσω ή το NULL για να εκτυπώσω όλες τις εγγραφές. Η **HT\_PrintBlockChain** είναι μια αναδρομική συνάρτηση που εκτυπώνει τις εγγραφές ξεκινώντας από αυτές του τελευταίου block εγγραφών της αλυσίδας πηγαίνοντας προς το πρώτο κάνοντας χρήση της **ΗΤ\_PrintRecord** η οποία εκτυπώνει το περιεχόμενο κάθε record με ένα συγκεκριμένο τρόπο.

Στη Makefile, για το hp έχει προστεθεί και αφαίρεση του data.db.