

**Υλοποίηση Συστημάτων Βάσεων Δεδομένων (Κ18)**  
**Χειμερινό Εξάμηνο 2018 - 2019**  
**Άσκηση 1/2**

**Ονοματοεπώνυμο και ΑΜ Φοιτητών :**

Κουτσοβασίλης Νικόλαος  
1115201500076  
Μακρυγεώργος Νικόλαος  
1115201500238

**Μεταγλώττιση και εκτέλεση :**

```
gcc -o main main.c ht_lib.c sht_functions.c BF_64.a
```

Παράγεται το εκτελέσιμο αρχείο main

**Παραδοτέα Αρχεία :**

BF.h	HT.h	sht_functions.h	Recordd.h
BF_64.a	ht_lib.c	sht_functions.c	main.c

BF.h , BF\_64.a : Τα αρχεία που μας δόθηκαν.

HT.h , ht\_lib.c : Συναρτήσεις για το πρωτεύων ευρετήριο.

sht\_functions.h , sht\_functions.c : Συναρτήσεις για το δευτερεύων ευρετήριο και συναρτήσεις στατιστικών.

Recordd.h : Περιέχει τα struct Record και SecondaryRecord.

main.c : main ελέγχου των διαδικασιών των HT και SHT

**Περιγραφή Προγράμματος :**

**HT.h :** Το αρχείο επικεφαλίδας περιέχει τα προτότυπα των συναρτήσεων για έλεγχο του ευρετηρίου, τον ορισμό του struct πληροφοριών που πρέπει να κρατάμε στο πρώτο block όπως δόθηκε από την εκφώνηση και το struct ορισμού του block για το πρωτεύων ευρετήριο. Κάθε block στο πρωτεύων ευρετήριο εκτός του πρώτου περιέχει 6 πλήρης εγγραφές, 1 ακέραιο για το πόσες εγγραφές έχει κάθε στιγμή το block και 1 ακέραιο που αναφέρεται στο επόμενο block.

**ht\_lib.c :** Περιέχει τις συναρτήσεις για έλεγχο του πρωτεύων ευρετηρίου καθώς και την συνάρτηση κατακερματισμού.

HashInt() :

Η συνάρτηση κατακερματισμού παίρνει ορίσματα το id της εγγραφής και το πλήθος των bucket και επιστρέφει έναν τυχαίο ακέραιο στο διάστημα από 0 μέχρι bucket-1.

HT\_CreateIndex() :

Δημιουργεί το αρχείο με όνομα filename και στη συνέχεια δεσμεύει bucket + 1 blocks. Στο πρώτο block γράφει την δομή με τις πληροφορίες του αρχείου. Τα υπόλοιπα blocks είναι τα πρώτα blocks

για κάθε bucket τα οποία αρχικά είναι άδεια. Η αντιστοίχιση των buckets με τα blocks είναι η ακόλουθη: Το πρώτο block με αριθμό 0 δεσμεύεται για την δομή πληροφορίας.

Αρα το bucket 0 αντιστοιχεί στο block με αριθμό 1  
το bucket 1 αντιστοιχεί στο block με αριθμό 2  
το bucket 2 αντιστοιχεί στο block με αριθμό 3  
κ.ο.κ.

Ο τρόπος που γίνεται αυτό είναι ο εξής : Δεσμεύουμε block με την BF, διαβάζουμε το block, γράφουμε στο block την πληροφορία που θέλουμε και ξαναγράφουμε το block πάλι πίσω.

HT\_InsertEntry() :

Αρχικά βρίσκει σε ποιο bucket ταιριάζει η εγγραφή με την συνάρτηση κατακερματισμού. Στη συνέχεια εφαρμόζει την επαναληπτική διαδικασία. Διαβάζει το πρώτο block του bucket και ελέγχει αν έχει χώρο. Αν υπάρχει χώρος γράφει την καινούργια εγγραφή και αυξάνει τον μετρητή που μας λέει πόσες εγγραφές έχει το block. Τέλος γράφει το block πάλι πίσω και επιστρέφει τον αριθμό του block. Αν δεν υπάρχει χώρος στο πρώτο block του bucket τότε κοιτάει αν έχει επόμενο και αν έχει εφαρμόζεται πάλι η παραπάνω διαδικασία για το επόμενο block.

Στην περίπτωση που διατρέξει όλη την λίστα των blocks και δεν βρει χώρο τότε δημιουργεί ένα καινούργιο block και εφαρμόζει την παραπάνω διαδικασία για το νέο block.

HT\_DeleteEntry() :

Αρχικά βρίσκει σε ποιο bucket ταιριάζει η εγγραφή με την συνάρτηση κατακερματισμού. Στη συνέχεια εφαρμόζει επαναληπτικά την διαδικασία. Διαβάζει το πρώτο block του bucket αν βρεί την εγγραφή με αυτό το id την διαγράφει. Αν δεν ήταν η τελευταία του block βάζει στη θέση της την τελευταία και μειώνει τον μετρητή εγγραφών. Αν δεν βρεί την εγγραφή με αυτό το id διαβάζει το επόμενο block αν έχει και εφαρμόζεται η παραπάνω διαδικασία.

getAllEntries() :

Αρχικά βρίσκει σε ποιο bucket ταιριάζει η εγγραφή με την συνάρτηση κατακερματισμού. Διαβάζει επαναληπτικά την λίστα του bucket με τα blocks. Αν βρεί την εγγραφή με το συγκεκριμένο id εκτυπώνει την εγγραφή και επιστρέφει τον αριθμό των blocks που διάβασε μέχρι να βρεί την εγγραφή. Επίσης προστίθεται και το αρχικό block που περιέχει την πληροφορία του αρχείου. Αν δεν βρεί την εγγραφή τότε επιστρέφει -1. Τέλος εκτυπώνονται πόσες εγγραφές βρέθηκαν με αυτό το id. Στο πρωτεύον θα είναι πάντα 1 επειδή η εγγραφή με κάθε id θα είναι μοναδική.

**sht\_functions.h** : Το αρχείο επικεφαλίδας περιέχει τα προτότυπα των συναρτήσεων για έλεγχο του δευτερεύοντος ευρετηρίου, τον ορισμό του struct πληροφοριών που πρέπει να κρατάμε στο πρώτο block όπως δόθηκε από την εκφώνηση και το struct ορισμού του block στο δευτερεύον ευρετήριο. Κάθε block στο δευτερεύον ευρετήριο εκτός του πρώτου περιέχει 10 εγγραφές, 1 ακέραιο για το πόσες εγγραφές έχει κάθε στιγμή το block και 1 ακέραιο που αναφέρεται στο επόμενο block. Οι εγγραφές για το δευτερεύον ευρετήριο περιέχουν το id της εγγραφής, το block\_id δηλαδή το block που υπάρχει η εγγραφή στο πρωτεύον ευρετήριο και ένα πίνακα χαρακτήρων 40 θέσεων όπου θα αποθηκευτεί το name, surname ή address.

**sht\_functions.c** : Περιέχει τις συναρτήσεις για έλεγχο του δευτερεύον ευρετηρίου καθώς και την συνάρτηση κατακερματισμού και την συνάρτηση στατιστικών.

HashString():

Συνάρτηση κατακερματισμού για string. Επιστρέφει ακέραιο.

SHT\_CreateIndex() :

Ακολουθείτε ίδια λογική όπως στο πρωτεύων.

SHT\_SecondaryInsertEntry() :

Ακολουθείτε ίδια λογική όπως στο πρωτεύων μόνο που πριν την εισαγωγή της εγγραφής βρίσκει σε ποίο πεδίο απο τα name,surname,address θα εφαρμοστεί η hash\_function με βάση τι ευρετήριο έχουμε δημιουργήσει.

SHT\_SecondaryGetAllEntries() :

Αρχικά βρίσκει σε ποίο πεδίο απο τα name,surname,address θα εφαρμοστεί η hash\_function με βάση τι ευρετήριο έχουμε δημιουργήσει.Διατρέχει όλη την λίστα των blocks του bucket και για κάθε εγγραφή που βρίσκει πηγαίνει στο block του πρωτεύων ευρετηρίου που υπάρχει η εγγραφή και κάνει αναζήτηση στο block μέχρι να βρεί την εγγραφή με αυτό το id.

Επιστρέφει το πλήθος των blocks που διάβασε μέχρι να βρεί όλες τις εγγραφές συν 1 για το πρώτο block πληροφορίας και εκτυπώνει πόσες εγγραφές βρέθηκαν.

HashStatistics() :

Ανάλογα με τι αρχείο HT ή SHT καλεί αντίστοιχα μια συνάρτηση.Κάθε συνάρτηση υπολογίζει και εκτυπώνει τα στατιστικά.

Οι συναρτήσεις Open\_Index τόσο για το πρωτεύων όσο και το δευτερεύων ελέγχουν αν το αρχείο είναι αρχείο κατακερματισμού και του τύπου τους δηλαδή HT ή SHT. Σε περίπτωση κάποιου λάθους επιστρέφουν NULL.

Καταλήγοντας έχουμε μια main ελέγχου των συναρτήσεων όπου δημιουργει ένα πρωτεύων και δύο δευτερεύοντα ευρετήρια και καλεί διάφορες συναρτήσεις ελέγχου για δοκιμή.

Στα αρχεία υπάρχουν αρκετά σχόλια όπου θεωρείται αναγκαίο. Επίσης οι συναρτήσεις κατακερματισμού είναι οι γνωστές universal hash functions όπου προτάθηκαν στο φροντιστήριο της άσκησης.