

Εργασία 2

Αλέξανδρος Γιαννούλας – 1115201800027

Γιώργος-Αχιλλέας Παπαγεωργίου – 1115201800147

ΕΚΤΕΛΕΣΗ

Στο directory SHT_lib αντίστοιχα:

- **make** : Παράγει το εκτελέσιμο 'main' το οποίο αποτελεί το demo κάθε δομής.
- **make run** : Παράγει το εκτελέσιμο 'main' και το εκτελεί.
- **make clean** : Διαγράφει όλα τα εκτελέσιμα και αντικειμενικά αρχεία που έχουν παραχθεί απο τη make.

ΓΕΝΙΚΑ

- Χρησιμοποιήθηκε η παράμετρος -std=c++11 για τη μεταγλώττιση.
- Χρησιμοποιήθηκε η παράμετρος -w για να αγνοηθούν τα warnings. Τα warnings αφορούν το εξής: Για να συμβαδίζουν οι ορισμοί την συναρτήσεων με αυτούς της εκφώνησης ορίσαμε τα arguments ως char* και όχι string, ενώ περνάμε μεταβλητές τύπου string.
- Η μεταβλητή BF_Errno του αρχείου BF.h έγινε extern για να αποφευχθεί το πρόβλημα πολλαπλών ορισμών.
- Το αρχείο 'records1K.txt' έχει τροποποιηθεί ώστε να υποδुकνύει καλύτερα τη λειτουργικότητα του secondary index. Έχουν προστεθεί εγγραφές με surname 'surname_10' και διαφορετικά id ώστε να είναι επιτυχής η εισαγωγή τους.
- Για οποιαδήποτε διευκρίνηση σχετικά με το HashTable βλ. README εργασίας 1.

SHT_LIB

- Με την κλήση της `SHT_CreateSecondaryIndex` δημιουργείται ένα αρχείο, το δευτερεύον ευρετήριο το οποίο αποτελείται από δείκτες σε blocks τα οποία περιέχουν SecondaryRecords. Η γραμμή n του αρχείου αυτού περιλαμβάνει τον δείκτη στο πρώτο block του bucket στο οποίο βρίσκεται η εγγραφή με $\text{hash}(\text{surname}) = n$.
- Η SecondaryRecord αποτελείται από ένα surname και ένα blockId που είναι το block στο οποίο υπάρχει αυτό το surname. Κατά την εισαγωγή ενός SecondaryRecord στο secondary index ελέγχουμε αν υπάρχει ήδη και αν υπάρχει δεν το εισάγουμε ξανά. Εφόσον κατά την αναζήτηση θα ελέγχσουμε όλο το block στο οποίο ξέρουμε ότι υπάρχει μια εγγραφή με το ζητούμενο surname, θα εμφανίσουμε και όλες τις άλλες εγγραφές με το ίδιο surname αυτού του block.
- Στη δομή SHT_Info κρατάμε τον αριθμό του block που την περιέχει ώστε να το προσπεράσουμε όταν έρθει η ώρα της αναζήτησης ή της εισαγωγής.

- Η `SHT_SecondaryInsertEntry` δουλεύει ακριβώς όπως η `HT_InsertEntry` κατά την εισαγωγή του Record. Μόλις γίνει η εισαγωγή, κρατάμε τον αριθμό block στο οποίο έγινε η εισαγωγή και μαζί με το surname της εγγραφής το προσθέτουμε στο secondary index. Η `SHT_SecondaryInsertEntry` θα μπορούσε να απλοποιηθεί πολύ αν καλούσαμε μέσα της την `HT_InsertEntry` τροποποιημένη ώστε να επιστρέφει τον αριθμό block στο οποίο έγινε η εισαγωγή. Για να γίνει όμως αυτό θα έπρεπε να αλλάξουμε το πρότυπο της συνάρτησης, πράγμα που δεν επιτρέπεται.
- Εφόσον όλες οι συναρτήσεις HT και SHT ήταν ξεχωριστές, θεωρήσαμε ότι και η hashStatistics θα έπρεπε να είναι διαφορετική ώστε να ακολουθεί το ίδιο μοντέλο, καθώς αυτή η υλοποίηση ήταν μία από τις επιλογές που μας δώσατε στις διευκρινήσεις.
- String Hash Function source: <http://www.cse.yorku.ca/~oz/hash.html>