

Ξανθοπούλου Μαρία - 1115201400300

Ευαγγελάτος Χαράλαμπος - 1115201400041

Υλοποίηση Συστημάτων Βάσεων Δεδομένων - Εργασία 1/2

A: Συναρτήσεις Hash Table

Υλοποιήθηκαν οι παρακάτω συναρτήσεις :

- ❖ *HT_CreateIndex* : Δημιουργεί και αρχικοποιεί ένα πρωτεύον αρχείο στατικού κατακερματισμού με όνομα fileName. Στο πρώτο block αποθηκεύουμε τα εξής: typeOfFile , attrType , attrName , attrLenght , number of buckets. Επίσης δημιουργούμε ένα ευρετήριο στο 2ο block , και κάνουμε allocate άλλα n blocks όπου $n = \text{number of buckets} - 1$ ένα αρχικό block για κάθε bucket (αν χρειαστούν overflow blocks θα γίνουν allocate απο το bucket που χρειάζεται εκείνη τη στιγμή).
- ❖ *HT_OpenIndex* : Φτιάχνει μια δομή τύπου HT info και τη γεμίζει με τις κατάλληλες πληροφορίες.
- ❖ *HT_CloseIndex* : Ελευθερώνει την προαναφερθήσα δομή και κλείνει το index.
- ❖ *HT_InsertEntry*: Γίνεται εισαγωγή μίας εγγραφής στο πρωτεύον αρχείο κατακερματισμού.Ελέγχει ποιο είναι το key attribute του δοθέντος record(πιθανότατα το int id αλλά έχει γίνει πρόβλεψη και οι HT functions λειτουργούν και με κάποιο απο τα άλλα πεδία ως primary key) και υπολογίζει το hash value του attribute αυτού. Έτσι ξέροντας σε ποιο bucket πρέπει να μπει, καλούμε την συνάρτηση insert_to_bucket η οποία βρίσκει το πρωτο block του bucket αυτου, ελέγχει αν η εγγραφή χωράει (αν χωράει την εισάγει) και αν δεν χωράει ακολουθεί το “link” του block προς το επόμενο block ελέγχει αν χωράει εκεί και αν όχι ακολουθεί το link κ.ο.κ. μέχρι να βρεί ένα block στο οποίο χωράει όπου και την εισάγει. Αν μετά την εισαγωγή το block δεν χωράει άλλες εγγραφές τότε κάνουμε allocate ένα νέο (block‘) και δίνουμε τον αριθμό αυτού ως link στο block (στο οποίο βρισκόμαστε τώρα)
- ❖ *HT_GetAllEntries*: Με βάση τη τιμή που μας δίνεται και ποιο είναι το key πεδίο, βρίσκουμε το hash value για να δούμε σε ποιο bucket βρίσκεται το record που μας ήρθε σαν όρισμα. Ψάχνοντας ένα ένα τα block του bucket αυτου και τα records αυτών βρίσκουμε το record μας. Εκτυπώνουμε αυτο το record.
- ❖ *HT_DeleteEntry*: Η συνάρτηση αυτή λειτουργεί με τον ίδιο τρόπο με την HT_GetAll-Entries απλά αντι να εκτυπώνει το record το αντικαθιστά με ένα “κενό” record.

B: Συναρτήσεις Secondary Hash Table

Υλοποιήθηκαν οι παρακάτω συναρτήσεις :

- ❖ *SHT_CreateSecondaryIndex* : Δημιουργεί και αρχικοποιεί ένα δεύτερο αρχείο στατικού κατακερματισμού με διαφορετικό όνομα από το πρωτεύον αρχείο (sfileName). Στο πρώτο block αποθηκεύουμε τα εξής: typeOfFile , attrName , attrLength , number of buckets , fileName . Επίσης δημιουργούμε ένα ευρετήριο στο 2ο block , και κάνουμε allocate αλλά n blocks όπου $n = \text{number of buckets} - \text{ένα αρχικό block για κάθε bucket (αν χρειαστούν overflow blocks θα γίνουν allocate από το bucket που χρειάζεται εκείνη τη στιγμή)}$.
- ❖ *SHT_OpenSecondaryIndex* : Φτιάχνει μια δομή τύπου SHT info και τη γεμίζει με τις κατάλληλες πληροφορίες.
- ❖ *SHT_CloseSecondaryIndex*: Ελευθερώνει την προαναφερθήσα δομή και κλείνει το index.
- ❖ *SHT_SecondaryInsertEntry*: Ελέγχει ποιο είναι το key attribute του δοθέντος record και υπολογίζει το hash value του attribute αυτού. Έτσι ξέροντας σε ποιο bucket καλούμε την συνάρτηση s_insert_to_bucket η οποία βρίσκει το πρώτο block του bucket αυτού, ελέγχει αν η εγγραφή χωράει (αν χωράει την εισάγει) και αν δεν χωράει ακολουθεί το “link” του block προς το επόμενο block ελέγχει αν χωράει εκεί και αν όχι ακολουθεί το link κ.ο.κ. μέχρι να βρεί ένα block στο οποίο χωράει όπου και την εισάγει. Αν μετά την εισαγωγή το block δεν χωράει άλλες εγγραφές τότε κάνουμε allocate ένα νέο (block‘) και δίνουμε τον αριθμό αυτού ως link στο block (που βρισκόμαστε τώρα)
- ❖ *SHT_SecondaryGetAllEntries*: Με βάση τη τιμή που μας δίνεται και ποιο είναι το key πεδίο βρίσκουμε το hash value για να δούμε σε ποιο bucket βρίσκεται το secondary record. Ψάχνοντας ένα ένα τα block του bucket και τα records αυτών βρίσκουμε το secondary record μας από το οποίο αντλούμε το σε ποιο block του primary index βρίσκεται η εγγραφή που μας ενδιαφέρει. Μπαίνουμε σε αυτό το block και ψάχνουμε την εγγραφή με τιμή του name – surname – address ίδια με αυτή που μας ήρθε σαν όρισμα. Εκτυπώνουμε αυτή την εγγραφή.

Γ: Πως να εκτελεστεί το πρόγραμμα :

Η makefile παρέχει 2 εκδοχές εκτελέσιμου :

- 1) `make custom` : χρησιμοποιεί τη δική μας main , μόλις κάνετε make εκτελείτε ως `./run file1.txt file2.txt` , όπου file1 και file2 αρχεία εγγραφών - έχουν επισυναφθεί τα records_named.txt και records_named2.txt με λίγες εγγραφές "πραγματικων" ονομάτων ώστε να μπορείτε εύκολα να ελέγξετε τα αποτελέσματα (φυσικά δουλεύει κανονικά το να δώσετε και τα records1K.txt, records5K.txt κλπ)
- 2) `make` : χρησιμοποιεί τη δοθείσα main , την εκτελείτε με τον γνωστο τρόπο `./run x y` , όπου x ένας int που εκφράζει πόσες εγγραφές θέλουμε να εισαχθούν, και y ένας δεκαδικός $0 \leq y \leq 1$ που εκφράζει πόσες εκ αυτών θέλουμε ύστερα να διαγραφούν από το primary index (μετά δημιουργείται και το secondary index και γίνονται εισαγωγές και αναζητήσεις και σε αυτό κανονικά).