Εργασία 2

Υλοποίηση Συστημάτων και Βάσεων Δεδομένων

Πρασιανάκης Στυλιανός :sdi1700233

Τζιεράκης Ιωάννης:sdi1700163

Παντελάκης Γεώργιος:sdi1900140

Γενικές Παραδοχές:

Ο πίνακας openfiles αποτελείται από integers που αντιστοιχούν σε filedesc, έτσι ώστε να ξέρουμε σε ποιο αρχείο δείχνει το κάθε index.

Στο πρώτο block κάθε αρχείου αποθηκεύουμε τις εξής πληροφορίες :

•Πληροφορία για το αν το αρχείο είναι hashfile

•Το depth του αρχείου ή καλύτερα τον αριθμό των buckets.

•Τον αριθμό των εγγραφών που χωράνε σε κάθε block

Στα επόμενα block έχουμε το ευρετήριο, δηλαδή κάθε bucket αντιστοιχεί στον αριθμό του block που δείχνει. Αν ο αριθμός είναι -1 τότε το bucket δε δείχνει σε κανένα block ακόμα. Στη συνέχεια βρίσκονται τα blocks που περιέχουν τις εγγραφές που εισάγουμε.

*Για να εκτελέσουμε το πρόγραμμα τρέχουμε ένα από τα 3 scripts main1.sh, main2.sh, main3.sh, με την εντολή bash μπροστά πχ: bash main1.sh

→ Τα scripts κάνουν και make.

→ Τα scripts είναι στο φάκελο code μαζί με τον κώδικα.

HT_Init():

Σε αυτή την συνάρτηση αρχικοποιούμε έναν πίνακα με όνομα openfiles μεγέθους MAX_OPEN_FILES όπου σε αυτό τον πίνακα αποθηκεύεται το αναγνωριστικό κάθε αρχείου. Στην αρχή όλες τις θέσεις τις αρχικοποιούμε με -1 για να δείξουμε ότι δεν έχει αποθηκευτει κάποιο ανοιχτό αρχείο.

HT_CreateIndex():

Σε αυτή τη συνάρτηση αρχικοποιούμε ένα αρχείο επεκτατού κατακερματισμού.

Ουσιαστικά δημιουργούμε ένα αρχείο με την συνάρτηση **BF_CreateFile** ,και το ανοίγουμε με την συνάρτηση **BF_OpenFile.**

Έπειτα παίρνουμε το πρώτο Block γράφουμε στην αρχή του Block (πρώτα 5 Bytes) το την λέξη «**hash**» για να ώστε να ξέρουμε ότι το αρχείο είναι hashfile,έπειτα γράφουμε το βάθος του ευρετηρίου και στο τέλος τον αριθμό των Records που χωράνε σε ένα block.

Έπειτα τα επόμενα block τα αρχικοποιούμε με -1 και κλείνουμε το αρχείο.

HT_OpenIndex():

Σε αυτή την συνάρτηση ανοίγει το αρχείο και εξετάζει εάν είναι αρχείο κατακερματισμού ελέγχοντας το πρώτο block αν περιέχει στα πρωτα 5 bytes την συμπβολοσειρά «**hash**».

Σε περίπτωση που είναι αρχείο κατακερματισμού πηγαίνει και βρίσκει στον πίνακα openfiles αν υπάρχει κενή θέση και καταχωρεί το αναγνωριστικό του αρχείου.

HT_Close_File():

Η Συνάρτηση αυτή ψάχνει το ένα ανοικτό αρχείο στον πίνακα open_files που είναι στην θέση indexDesc.Αν σε στην θέση αυτή τη θέση εμπεριέχεται η τιμή -1 σημαίνει πως δεν υπάρχει ανοικτό αρχειο σε αυτή τη θέση του πίνακα.

Αν υπάρχει καλείται η BF_CloseFile και κλείνει το αρχείο και μετά ελευθερώνουμε την θέση από τον πίνακα βάζωντας σαν περιεχόμενο σε αυτήν τη θέση την τιμή -1.

HT_InsertEntry():

Σε αυτή τη συνάρτηση ψάχνουμε ένα ανοικτό αρχείο στον πίνακα open_files που είναι στην θέση indexDesc. Αν υπάρχει τότε με βάση το απλό μας hashcode τσεκάρουμε το πρώτο bucket και βάζουμε εκεί ένα καινούριο block και κάνουμε τις κατάλληλες διαδικασίες για να ετοιμάσουμε το bucket και το επόμενο block. Αν το bucket είναι γεμάτο πάμε και παίρνουμε το 1ο block της αλυσίδας.

Μετά από αυτή τη διαδικασία ψάχνουμε το τελευταίο block της αλυσίδας. Αν είναι γεμάτο τότε προσθέτουμε ένα block στην αλυσίδα. Στο τέλος της συνάρτησης αποθηκεύουμε το record (ή entry) στην τελευταία θέση του block που έχουμε επιλέξει ανάλογα με την παραπάνω διαδικασία και αυξάνουμε κατά ένα το number_of_blocks.

HT_PrintAllEntries():

Σε αυτή τη συνάρτηση εμφανίζουμε όλα τα entries ανάλογα με το id που δίνεται όταν καλείται η συνάρτηση, αφού ψάξουμε ένα ανοικτό αρχείο στον πίνακα open_files που είναι στην θέση indexDesc.

→ Αν το id είναι **NULL** τότε εμφανίζουμε όλα τα entries:

Σε αυτή την περίπτωση παίρνουμε όλα τα hashblocks και μετά για κάθε bucket εκτυπώνουμε τα records με τις κατάλληλες επαναλήψεις.

→ Αν το id είναι integer τότε η συνάρτηση ψάχνει να βρει αν υπάρχει record με το id που δώθηκε και αν ναι το εκτυπώνει.