Πληροφορική & Τηλεπικοινωνίες
Υλοποίηση Συστημάτων Βάσεων Δεδομένων
Χειμερινό Εξάμηνο 2020 – 2021

Γιώργος – Σάββας Δημητρίου 1115201800045
Χριστίνα Διαμαντή 1115201800046

- Άσκηση 2 -

Η παρούσα εργασία αφορά την δημιουργία ενός δευτερεύοντος ευρετηρίου στατικού κατακερματισμού (Secondary Hash Table) πάνω στο χαρακτηριστικό surname. Το παραπάνω ευρετήριο βασίζεται στο πρωτεύον ευρετήριο στατικού κατακερματισμού το οποίο υλοποιήθηκε στην άσκηση 1. Για την δημιουργία του δευτερεύοντος ευρετηρίου υλοποιήθηκαν οι παρακάτω συναρτήσεις.

SHT_CreateSecondaryIndex: Δημιουργεί ένα αρχείο δευτερεύοντος ευρετηρίου (Secondary Hash Table). Δεσμεύει ένα μπλοκ στο οποίο αφού το διαβάσει αποθηκεύει τα ακόλουθα δεδομένα:

1. Η συμβολοσειρά SecondaryHash
2. Το μήκος του κλειδιού
3. Το όνομα του κλειδιού
4. Τον αριθμό των κουβάδων
5. Το μήκος του ονόματος του αρχείου πρωτεύοντος ευρετηρίου
6. Το όνομα του αρχείου πρωτεύοντος ευρετηρίου

Έπειτα δεσμεύει τόσα μπλοκς όσος και ο αριθμός των κουβάδων. Τέλος αρχικοποιεί αυτά τα μπλοκς, γράφοντας το μηδέν, το οποίο συμβολίζει τον αριθμό των εγγραφών στο μπλοκ, στα πρώτα τέσσερα bytes. Στα επόμενα τέσσερα αποθηκεύουμε το μπλοκ ID του μπλοκ που συνδέεται με το προηγούμενο. Αυτό είναι επίσης αρχικοποιημένο με την τιμή μηδέν, που σημαίνει από συμβιβασμό ότι δεν υπάρχει επόμενο μπλοκ.

SHT_OpenSecondaryIndex: Ανοίγει ένα αρχείο δευτερεύοντος ευρετηρίου και διαβάζει το πρώτο μπλοκ του. Στη συνέχεια ανακτά τη συμβολοσειρά που αποθηκεύσαμε κατά την δημιουργία του αρχείου. Αν αυτή είναι διαφορετική της “SecondaryHash”, εκτυπώνει μήνυμα λάθους και επιστρέφει NULL. Διαφορετικά δεσμεύει μνήμη στο σωρό, στην οποία αποθηκεύει τις πληροφορίες του κλειδιού, των αριθμό των κουβάδων και το όνομα του πρωτεύοντος ευρετηρίου καθώς και το file descriptor του αρχείου. Όλα αυτά τα δεδομένα είναι αποθηκευμένα σε μια δομή SHT_info, η οποία επιστρέφεται.

SHT_CloseSecondaryIndex: Κλείνει ένα αρχείο δευτερεύοντος ευρετηρίου και αποδεσμεύει την μνήμη που δεσμεύτηκε για τις πληροφορίες του αρχείου.

SHT_SecondaryInsertEntry: Αρχικά είναι σημαντικό να αναφερθεί ότι κάθε φορά που καλείται η συγκεκριμένη συνάρτηση εισάγεται στο δευτερεύον ευρετήριο μία ειδική εγγραφή που περιέχει δύο ακέραιους. Ο ένας δείχνει το μπλοκ ID του πρωτεύοντος ευρετηρίου στο οποίο είναι αποθηκευμένη η εγγραφή που δίνεται ως όρισμα, ενώ ο δεύτερος δείχνει την θέση της στο μπλοκ. Αφού ανακτήσουμε το δεδομένα αυτά κατακερματίζουμε το surname της εγγραφής του ορίσματος. Διαβάζουμε το μπλοκ με ID το αποτέλεσμα του κατακερματισμού. Μετά ανακτούμε το ID του μπλοκ σύνδεσης. Όσο αυτό είναι διάφορο του 0, διαβάζουμε το μπλοκ σύνδεσης και κάνουμε το ίδιο. Έτσι φτάνουμε στο τελευταίο μπλοκ του κουβά και ελέγχουμε εάν χωράει ακόμα μια εγγραφή. Σε περίπτωση που χωράει, την προσθέτουμε στο τελευταίο μπλοκ. Αντίθετα δεσμεύουμε νέο μπλοκ , αποθηκεύουμε εκεί την εγγραφή και συνδέουμε το νέο μπλοκ με το προηγούμενο που αναφέραμε. Πλέον το νέο μπλοκ είναι το τελευταίο του κουβά. Σε κάθε περίπτωση, ενημερώνουμε κατάλληλα την μεταβλητή που αποθηκεύει το πλήθος των εγγραφών.

SHT_SecondaryGetAllEntries: Εδώ διακρίνουμε δύο περιπτώσεις. Εάν το κλειδί είναι NULL ή όχι. Στην πρώτη περίπτωση, καλείται η **HT_GetAllEntries**, εκτυπώνονται όλες οι εγγραφές και επιστρέφεται ο αριθμός των μπλοκ που διαβάστηκαν στο πρωτεύον ευρετήριο. Στη δεύτερη περίπτωση, κατακερματίζουμε το κλειδί value και διατρέχουμε ένα-ένα τα μπλοκς του κουβά στον οποίο οδηγούμαστε μέσω του κατακερματισμού. Αν υπάρχει κάποια εγγραφή της οποίας το surname να ταυτίζεται με το κλειδί, την εκτυπώνουμε. Ο τρόπος με τον οποίο ανακτούμε την πραγματική εγγραφή μέσω της ειδικής εγγραφής που έχουμε είναι ο εξής:

Διαβάζουμε το μπλοκ του πρωτεύοντος ευρετηρίου με μπλοκ ID ίσο με την τιμή του πρώτου ακέραιου της ειδικής εγγραφής. Έπειτα από αυτό το μπλοκ λαμβάνουμε την εγγραφή που βρίσκεται στη θέση που δηλώνει ο δεύτερος ακέραιος της ειδικής εγγραφής. Έτσι έχουμε την πραγματική εγγραφή και συγκρίνουμε το surname της με το value που έχουμε. Τελικά επιστρέφουμε τον αριθμό των μπλοκς που διαβάσαμε στο δευτερεύον ευρετήριο.

SecondaryHashStatistics: Υλοποιήσαμε δύο βοηθητικές συναρτήσεις ώστε να μπορούμε να βρού με τα ζητούμενα με μεγαλύτερη ευκολία. Η numRecords, η οποία επιστρέφει το πλήθος των εγγραφών σε ένα κουβά, και η blocksNumber η οποία βρίσκει το πλήθος των μπλοκς σε έναν κουβά. Οι παραπάνω συναρτήσεις χρησιμοποιούν την μέθοδο της αναδρομής. Χρησιμοποιώντας αυτές, η SecondaryHashStatistics βρίσκει και τυπώνει τα ζη-

τούμενα αποτελέσματα. Στην πραγματικότητα πρόκειται για την ίδια HashStatistics της άσκησης 1, με την μόνη διαφορά ότι εκτελεί τις ίδιες λειτουργίες για το δευτερεύον ευρετήριο.

Μεταγλώττιση και εκτέλεση του προγράμματος

Η μεταγλώττιση και η εκτέλεση γίνεται με πολύ απλό τρόπο. Δημιουργήσαμε ένα Makefile το οποίο περιέχει τις απαραίτητες εντολές για την μεταγλώττιση του προγράμματος. Επίσης, υπάρχει ένα σύνολο από εντολές για την εκτέλεση του προγράμματος, για όλα τα αρχεία εισόδου που έχουν δοθεί με αρχείο σωρού, με αρχείο κατακερματισμού και με αρχείο δευτερεύοντος ευρετηρίου. Για την μεταγλώττιση του προγράμματος, πατάμε την εντολή make.

Για την εκτέλεση του προγράμματος υπάρχουν οι ακόλουθες εντολές:

make run_heap_1K
make run_hash_1K
make run_+sht_1K
make run_heap_5K
make run_hash_5K
make run_+sht_5K
make run_heap_10K
make run_hash_10K
make run_+sht_10K
make run_heap_15K
make run_hash_15K
make run_+sht_15K

Για την εκκαθάριση των προσωρινών αρχείων πατάμε την εντολή make clean.

Το πρόγραμμα τρέχει με έναν προκαθορισμένο τρόπο ως προς τα ορίσματα που δέχεται μέσω της γραμμής εντολών. Σε περίπτωση που οι περιορισμοί δεν τηρούνται, το πρόγραμμα εκτυπώνει σχετικό μήνυμα που περιγράφει τον σωστό τρόπο εκτέλεσης και τερματίζει. Οι παραπάνω εντολές που δίνονται μέσω του Makefile τηρούν τους περιορισμούς αυτούς, ώστε το πρόγραμμα να εκτελείται επιτυχώς.