Προγραμματισμός συστήματος (K24)
Περίοδος 2018 – 2019
1η Εργασία

Νικόλαος Μακρυγεώργος
1115201500238

Μεταγλώττιση και Εκτέλεση :
Η μεταγλώττιση γίνεται μέσω του makefile γράφοντας make.
Διαγραφή εκτελέσιμου και αντικειμενικών αρχείων γράφοντας make clean.

Η εκτέλεση γίνεται όπως στην εκφώνηση δηλαδή
/bitcoin -a bitCoinBalancesFile -t transactionFile -v bitCoinValue -h1 senderHashtableNumOfEntries -h2 receiverHashtableNumOfEntries -b bucketSize

Τα ορίσματα μπορούν να δοθούν με οποιαδήποτε σειρά.
π.χ.
/bitcoin -b bucketSize -h2 receiverHashtableNumOfEntries -h1 senderHashtableNumOfEntries -v bitCoinValue -a bitCoinBalancesFile -t transactionFile

Παραδοτέα Αρχεία :
HashTable.c : Περιέχει τις δομές για τα Hash Table.
HashTable.h
my_struct.c : Η δικιά μου δομή για τις πληροφορίες καθε user.
my_struct.h
tree.c : Περιέχει τις δομές για το δένδρο.
tree.h
bitcoin_functions.c : Περιέχει βοηθητικές συναρτήσεις
bitcoin_functions.h
bitcoin.c : Περιέχει την main
Readme
makefile

Περιγραφή Προγράμματος :
Όλες οι δομές έχουν υλοποιηθεί απο εμένα.Από το διαδίκτυο έχω δεί μόνο τον τρόπο που παίρνω την τρέχουσα ημερομηνία και ώρα.
Ιδέες για τις δομές έχω πάρει απο παλαιότερες εργασίες στο μάθημα των δομών δεδομένων αλλά και άλλων μαθημάτων.
Η συνάρτηση κατακερματισμού Universal hash function for string την έχω πάρει απο εργασία μου στην Υλοποίηση βάσεων δεδομένων όπου η ιδέα της προερχόταν απο το διαδίκτυο.
Τέλος μερικές συναρτήσεις της βιβλιοθήκης stdio.h και stdlib.h όπως η strtok που δεν είχα ξαναχρησιμοποιήση είδα την λειτουργία τους απο το διαδίκτυο.

Η εργασία έχει υλοποιηθεί,μεταγλωττιστεί και εκτελεστεί αποκλειστικά στα μηχανήματα linux της σχολής.

Γενικά η υλοποίηση έχει γίνει αυστηρά με βάση την εκφώνηση και τις αναφορές στο piazza.

Η μόνη διαφορά σε σχέση με την εκφώνηση είναι οτι :

Ο δείκτης δεν είναι απο το hash table προς το δένδρο αλλά απο το δένδρο στο hast table.Το οποίο είναι επιτρεπτό σύμφωνα με αναφορά στο piazza.

Οι εντολές μπορούν να δουθούν όπως περιγράφεται παρακάτω.

/requestTransaction name1 name2 amount 1-1-2010 10:10
/requestTransaction name1 name2 amount
Μπορείς να δώσεις η να μην δώσεις ημερομηνία και ώρα.

/requestTransactions name1 name2 amount 1-1-2010 10:10;name3 name4 amaunt 2-2-2010 10:20;
Όσες συναλλαγές θέλουμε με η χωρίς ημερομηνία.Θέλει πάντα στο τέλος κάθε συναλλαγής ερωτηματικό.

/requestTransactions file
Δίνοντας ένα αρχείο όπου οι συναλλαγές είναι με την μορφή σαν να μην δίνουμε αρχείο δηλαδή σαν το παραπάνω.
Οι συναλλαγές και στις δυο περιπώσεις πρέπει να είναι συνεχόμενες.

/findEarnings name
Για να εκτυπώσει πληροφορίες απο όλες τις συναλλαγές.

/findEarnings name time1 date1 time2 date2
Για να επιλέξει τις συναλλαγές ανάμεσα απο αυτές τις ημερομηνίες και ώρες.

/findEarnings name time1 time2
Για να επιλέξει τις συναλλαγές με χρόνο ανάμεσα απο το time1 και time2

Αντίστοιχα για /findPayments

/walletStatus name : Δίνεις μόνο όνομα
/bitCoinStatus : Δίνεις το bitcoin_id
/traceCoin : Δίνεις το bitcoin_id
/exit : Καλεί συνάρτηση και διαγράφει όλη την μνήμη που δέσμευσε
Οι εντολές αυτές δίνονται στην main.

hash Table :

Έχω φτιάξει έναν πίνακα δεικτών μεγέθους όσο δώσεις στην γραμμή εντολης,όπου κάθε δείκτης δείχνει στο αρχκό κενό bucket.Το κάθε bucket είναι ενα struct που περιέχει εναν δείκτη για το επόμενο bucket αν χρειασθεί,έναν ακέραιο που μας λέει πόσες κενές θέσεις έχει το bucket και έναν δείκτη σε εγγραφές διότι δεν ξέρουμε απο την αρχή το πλήθος των εγγραφώ του bucket.
Κάθε εγραφή είναι ενα struct που περιέχει το user_id και έναν δείκτη για την λίστα των συναλλαγών που έχει κάνει ο χρήστης.
Καθε κόμβος που αναπαριστά μια συναλλαγή περιέχει τον δέκτη / αποστολέα και τα στοιχεία της συναλλαγής όπως το ποσό,την ημερομηνία και την ώρα.

Η συνάρτηση κατακερματισμού παίρνει ως όριμσα το user_id και το μέγεθος του hash table και επιστρέφει έναν αριθμό στο διάστημα [0, μέγεθος_hashTable).

my_struct :

Έχω επιλέξει να αποθηκεύω τους χρήστες και τα bitcoin που έχουν με έναν πίνακα απο λίστες με κόμβο κεφαλή.Κάθε κόμβος κεφαλή περιέχει το user_id,το συνολικό ποσό που έχει ο user_id. Επίσης περιέχει το μέγεθος της λίστας απο bitcoin και δύο δείκτες για την αρχή και τέλος της λίστας με τα bitcoin.Το μέγεθος και οι δείκτες με βοηθούν στο να ενημερώνω σωστά την λίστα. Κάθε κόμβος στη λίστα με τα bitcoin περιέχει το coin_id, το ποσό που έχει αυτό το bitcoin και δείκτες προηγούμενου και επόμενου κόμβου.

Tree :

Κάθε κόμβος του δένδρου περιέχει το χρήστη που έχει μέρος του bitcoin και το ποσό που έχει.Επίσης δύο δείκτες για τα hash tables.Τα bitcoin_id βρίσκονται σε έναν παράλληλο πίνακα ακεραίων.Δηλαδή το bitcoin που είναι στη θέση 5 του πίνακα υπάρχει στην 5 θέση του πίνακα απο δείκτες σε δένδρα.
Έχω ακολουθήσει την λογική του σχήματος της εκφώνησης.
Έπειδή είναι αρκετά περίεργο το να γράψω σχολαστικά πως υλοποίησα να δουλεύει το δένδρο θα τα αναφέρω ακριβώς στην εξέταση.

Για την λειτουργικότητα των συναρτήσεων των δομών έχω σχόλια πάνω απο κάθε συνάρτηση.

main :

Αρχικά η main καλεί κάποιες συναρτήσεις για να πάρει τα ορίσματα που δόθηκαν στην γραμμή εντολή,να ελέγξει αν το αρχείο με τους χρήστες είναι έγκυρο δηλαδή να περιέχει κάθε χρήστη και κάθε bitcoin μία φορα.Ελέγχει το αρχείο με τις συναλλάγες οτι υπάρχει κάθε transaction_id μόνο μια φορά.
Στην περίπτωση που κατι απο αυτά δεν είναι σωστά τερματίζει με μήνυμα λάθους.
Υπάρχουν και άλλοι έλεγχει κατα την διάρκεια των συναλλαγών όπως αν ο αποστολέας έχει χρήματα,αν ο αποστολέας είναι ίδιος με τον παραλήπτη,αν χρήστες σε μια συναλλαγή υπάρχουν στο wallet, αν η ημερομηνία μιας συναλλαγής είναι μεταγενέστερη της προηγούμενης συναλλαγής.
Όταν κατι δεν πάει καλά απο τα παραπάνω απλά ακυρώνεται η συναλλάγή,εμφανίζει κατάλληλο μήνυμα και προχωράει για την επόμενη.
Ύπάρχουν και ελέγχει όταν δίνονται οι εντολές για να μην τερματίσει ποτέ με segmation fault σε μια λάθος εντολή.
Στις περιπτώσεις που χρειάζεται ημερομηνία και ώρα αν δεν δοθούν απο τον χρήστη της εφαρμογής το πρόγραμμα πάιρνει την τρέχουσα ημερομηνία και ώρα.
Με την εντολή /exit καλείται συνάρτηση που διαγράφει όλη την μνήμη που είχε δεσμευτεί.
Κατά το τέλος του προγράμματος αποδεσμεύεται όλη μνήμη.

Καταλήγοντας υπάχουν σχόλια πάνω απο κάθε συνάρτηση αλλά και δίπλα απο εντολές όπου χρειάζεται.

Περισσότερες λεπτομέρειες στην προφορική εξέταση.