ΕΡΓΑΣΙΑ #1 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ
Group : APTIOI

A.M. : 1115201200106
Όνομα : Βασίλειος
Επίθετο : Μαυρομμάτης

ΠΑΡΟΥΣΙΑΣΗ ΚΩΔΙΚΑ

Η εργασία είναι υλοποιημένη σε C++.
Όλες οι δομές που έχουν υλοποιηθεί είναι δικές μου από εργασίες μαθημάτων κυρίως Αντικειμενοστραφή Προγραμματισμού, Λειτουργικών Συστημάτων και Δομών Δεδομένων με μικροαλλαγές για βελτιστοποίηση στο ζητούμενο.

Η εργασία αποτελείται από τα εξής **14** αρχεία:
1 αρχείο **Driver.cpp** το οποίο περιέχει την main του προγράμματος και αποτελεί τον κορμό της εργασίας.
2 αρχεία **HashTable.cpp** και το αντίστοιχο header **HashTable.hpp** που υλοποιούν τους πίνακες καταρκεματισμού τις λίστες των buckets και τα ίδια τα buckets.
2 αρχεία **TransactionsList.cpp** και το αντίστοιχο header **TransactionsList.hpp** που υλοποιούν τις λίστες και τα ίδια τα transactions του κάθε χρήστη.
2 αρχεία **BlockchainTree.cpp** και το αντίστοιχο header **BlockchainTree.hpp** που υλοποιούν τα BST όλων των BTC που κατέχουν οι χρήστες.
2 αρχεία **Wallet.cpp** και το αντίστοιχο header **Wallet.hpp** που υλοποιούν τα πορτοφόλια των χρηστών που συμμετέχουν.
2 αρχεία **TimeFrame.cpp** και το αντίστοιχο header **TimeFrame.hpp** που υλοποιούν τις ημερομηνίες των συναλλαγών σε αναπαράσταση ακεραίων.
2 αρχεία **Helpers.cpp** και το αντίστοιχο header **Helpers.hpp** που έχουν βοηθητικές δομές και συναρτήσεις ελέγχου εγκυρότητας δεδομένων τόσο για το input του χρήστη όσο και για τον έλεγχο των συναλλαγών από αρχικό αρχείο.
1 αρχείο **Makefile** για την δημιουργία του τελικού εκτελέσιμου προγράμματος : **bitcoin**

Αρχικά η main του προγράμματος ελέγχει οτι τα αρχικά arguments που απαιτούνται για την αρχικοποίηση των δομών (και μπορεί να έρθουν με οποιδήποτε σειρά) είναι έγκυρα και αν ναι προχωράει.
Αρχικοποιονται πρώτα τα wallets των χρηστών από το balanceFile με τους απαραίτητους ελέγχους μοναδικότητας BTC Ids και User Ids και στην συνέχεια επεξεργάζεται το αρχείο με τα αρχικά transactions απ’το οποίο γραμμή γραμμή ελέγχεται η εκυρότητα του εκάστοτε transaction και αν κριθεί valid ενημερωνονται οι δομές αλλίως απορρίπτεται και προχωράμε στην επόμενη συναλλαγή.

Αφού έχουν αρχικοποιηθεί οι δομές σωστά, το πρόγραμμα μπαίνει σε έναν ατέρμονο βρόγχο όπου περιμένει user input για τις εντολές που ζητήθηκαν για καινούρια transactions ή για προσπέλαση δεδομενών στις υπάρχοντες δομές.
Ελέγχεται κάθε εντολή και αν είναι valid εκτελλείται, διαφορετικά παρουσιάζεται μήνυμα λάθους και ο χρήστης να ξαναπροσπαθήσει αν θέλει, μέχρι ο χρήστης να ζητήσει να σταματήσει το πρόγραμμα δίνωντας την εντολή /exit
Έχω υλοποιήσει μία παραπάνω εντολή την /help με την οποία εμφανίζονται οι εντολές που μπορεί να δεχθεί το πρόγραμμα καθώς και ο τρόπος σύνταξης τους.
Τέλος όταν δωθεί η /exit το πρόγραμμα τελείωνει αφού έχει ελευθερωθεί όλη η μνήμη που δεσμεύθηκε δυναμικά για την συντήρηση των απαραίτητων δομών.

COMPILING + EXECUTION:

$ make
$ ./bitcoin -a bitCoinBalancesFile –t transactionsFile -v bitCoinValue -h1 senderHashtableNumOfEntries –h2 receiverHashtableNumOfEntries –b bucketSize (με οποιαδήποτε σειρά αρκεί να υπάρχουν όλα)
$ make clean

ΥΛΟΠΟΙΗΘΗΚΕ ΣΕ:

Language: C++(version 11)
OS: Ubuntu 18.04.02 LTS
Text Editor: Sublime Text 3
Debugger: Valgrind 3.13.0