

Κ24: Προγραμματισμός Συστήματος

Ονοματεπώνυμο: Κωνσταντίνος Πασχόπουλος

Αριθμός Μητρώου: 1115201500127

Η εργασία μου αποτελείται από τα εξής αρχεία:

Στο `mytypes.h` βρίσκονται όλοι οι τύποι δεδομένων που χρησιμοποιώ. Στο `bitcoin.c` βρίσκεται η `main` της εργασίας μου. Στο `bitcoin_functions.c` βρίσκονται όλες οι συναρτήσεις που χρησιμοποιώ για να διαβάσω τα αρχεία `transactionsFile` και `bitCoinBalancesFile` και όλες οι συναρτήσεις που χρησιμοποιώ για να δημιουργήσω τις δομές δεδομένων που χρειάζονται. Στο `command_functions.c` υπάρχουν οι συναρτήσεις που υλοποιούν τις διάφορες εντολές που μπορεί να δώσει ο χρήστης.

Ο δομές που χρησιμοποίησα είναι:

Μια συνδεδεμένη λίστα (`List`) στην οποία σε κάθε κόμβο αποθηκεύω ένα `bitcoin` (`bitcoin_node`). Ο κάθε κόμβος έχει το `bitCoinID` και ένα δείκτη στη ρίζα του δέντρου του συγκεκριμένου `bitcoin`.

Μια λίστα (`wallet`) στην οποία σε κάθε κόμβο αποθηκεύω το `wallet` κάποιου χρήστη (`wallet_node`). Ο κάθε κόμβος έχει το `walletID` του `wallet` και μια λίστα από όλα τα `bitcoins` που του ανήκουν (`leaf`). Ο κάθε κόμβος αυτής της λίστας έχει ένα δείκτη στο `bitcoin` που του ανήκει και ένα δείκτη σε κάποιο κόμβο του δέντρου του `bitcoin`.

Το δέντρο του κάθε `bitcoin` αποτελείται από κόμβους (`tree_node`) μέσα στους οποίους βρίσκονται το `walletID` στο οποίο ανήκει αυτός ο κόμβος του `bitcoin`, η αξία αυτού του κόμβου και τα στοιχεία του συναλλαγής στην οποία συμμετείχε αυτός ο κόμβος (`transaction_info`).

Το κάθε `hash table` έχει έναν πίνακα από δείκτες σε `bucket`. Το κάθε `bucket` έχει έναν πίνακα από `bucket_cell` (`entries`) και έναν δείκτη σε επόμενο `bucket` αν χρειαστεί να δημιουργηθούν `overflow buckets`. Κάθε κελί του πίνακα `entries` χρησιμοποιεί την μεταβλητή `empty` για να δείξει αν είναι άδειο ή όχι, το `walletID` που είναι το κλειδί και μια λίστα από συναλλαγές. Ο κάθε κόμβος αυτής της λίστας έχει έναν δείκτη `next` που δείχνει στο επόμενο στοιχείο της λίστας, έναν δείκτη `tree` που δείχνει στον κόμβο του δέντρου που έλαβε μέρος στην συναλλαγή (και περιέχει τα στοιχεία της συναλλαγής) και έναν δείκτη `next_bitcoin` που είναι μια λίστα από τα υπόλοιπα `bitcoins` που πιθανώς χρησιμοποιήθηκαν στην ίδια συναλλαγή.

Για να κάνω πιο εύκολες τις συγκρίσεις μεταξύ ημερομηνιών και ωρών όπου χρειάζεται τα μετατρέπω σε Unix Time με την συνάρτηση `getUnixTime`.

Στα hash tables υπάρχουν οι μεταβλητές `greatestTransactionID` και `latestTransaction` τις οποίες χρησιμοποιώ για να βρίσκω γρήγορα ένα καινούργιο μοναδικό `transactionID` και για να ελέγχω αν μια συναλλαγή είναι μεταγενέστερη της τελευταίας συναλλαγής που έχει καταγραφεί αντίστοιχα.

Κατά το parsing των αρχείων `transactionsFile` και `bitCoinBalancesFile` στις περιπτώσεις που δύο χρήστες έχουν το ίδιο ID ή τους ανήκει το ίδιο bitcoin ή κάποιο `transactionID` δεν είναι μοναδικό καλείτε η συνάρτηση `exitProgram` η οποία αποδεσμεύει όση μνήμη χρησιμοποιήθηκε και τερματίζει.

Όταν ο χρήστης θέλει να σταματήσει να δίνει συναλλαγές στην πρώτη συνάρτηση `requestTransactions` πρέπει να γράψει `exit` και να πατήσει `enter` για να επιστρέψει στην κανονική ροή του προγράμματος.

Για να γίνει compilation υπάρχει το Makefile το οποίο κάνει separate compilation.