

# Προγραμματισμός Συστήματος – Τμήμα Περιττών 2018/2019

## Εργασία 1

ΜΑΡΑΓΚΟΖΑΚΗΣ ΓΕΩΡΓΙΟΣ – 1115201500089

- **Δομή Αρχείων:** bitcoin.c, structures.c, structure.h, functions.c, functions.h, Makefile
- **Δομές Δεδομένων:**
  - 2 Hash Table (sender και receiver)
  - Bucket δεικτών σε Wallets και Bucket List
  - Wallet για κάθε χρήστη, με δείκτες προς λίστα των συναλλαγών του και προς λίστα των bitcoin που έχει στο πορτοφόλι του
  - Transaction και Transaction List
  - Bitcoin και Bitcoin List
  - Bitcoin Tree
- **Συναρτήσεις Δομών Δεδομένων:** structures.c
  - **create, delete, print, find:** Οι συναρτήσεις αυτές είναι απλές ως προς την λειτουργεί που επιτελούν και η υλοποίηση τους βασίστηκε σε υλοποιήσεις μου για εργασίες στο μάθημα των Λειτουργικών Συστημάτων το Χειμερινό Εξάμηνο του ακαδημαϊκού έτος 2018-2019. Παρακάτω, επεξηγούνται οι υλοποιήσεις ορισμένων συναρτήσεων οι οποίες χρήζουν διευκρινήσεων.
  - **hash\_function:** Η υλοποίηση της hash function για την πρόσβαση στις θέσεις του hash table έγινε με την ιδέα του αθροίσματος των ascii κωδικών όλων των στοιχείων της εκάστοτε συμβολοσειράς, πολλαπλασιασμένα το καθένα από διαφορετική δύναμη ενός πρώτου αριθμού ώστε να επιτευχθεί “μοναδικότητα” στο τελικό άθροισμα. Η ιδέα για αυτήν την υλοποίηση επηρεάστηκε από [εδώ](#).
  - **create\_Hash\_Table:** Δημιουργείται ένας δυναμικός πίνακας δεικτών σε λίστες από Buckets. Επιπλέον, κάθε Hash Table περιλαμβάνει ένα πεδίο που αφορά το μέγεθος του Bucket του.
  - **create\_Bucket :** Δημιουργείται ένας πίνακας δεικτών σε Wallets το μέγεθος του οποίου είναι bucketsize(αριθμός bytes) / sizeof(Bucket). Το ακέραιο πηλίκο της διαίρεσης είναι οι θέσεις του πίνακα δεικτών, των οποίων ο αριθμός τους επί το μέγεθος bytes της κάθε θέσης μπορεί να είναι μικρότερο ή ίσο του bucketsize.
- **Συναρτήσεις υλοποίησης λειτουργιών και ερωτημάτων:** functions.c
  - **generate\_transaction\_id :** Λαμβάνεται ως όρισμα το μεγαλύτερο transaction id που δημιουργήθηκε προηγουμένως και εάν το μέγεθος του είναι μικρότερο από το όριο, 15 χαρακτήρες τότε του προστίθεται ο χαρακτήρας 0. Εάν έχει εξαντλήσει το όριο χαρακτήρων τότε από δεξιά προς τα αριστερά αυξάνεται κατά ένα ο πρώτος χαρακτήρας που ικανοποιεί τη συνθήκη να είναι 0-9, a-z ή A-Z.
  - **read\_from\_file :** Υλοποιείται η διαδικασία διαβάσματος του αρχείου bitCoinBalancesFile και η εισαγωγή των χρηστών και των bitcoin του καθενός στην εφαρμογή. Γίνεται έλεγχος για διπλότυπους χρήστες και bitcoins και αντίστοιχα απορρίπτονται. Μια ιδιαιτερότητα είναι ότι από αυτό το στάδιο εισάγονται όλοι οι χρήστες στους δύο Hash Tables, sender και receiver ώστε να έχουμε άμεση πρόσβαση σε όλους και να εξοικονομήσουμε χώρο, από το να δημιουργούνταν νέα δομή. Επιπλέον, για τα bitcoin χρησιμοποιείται μια βοηθητική Λίστα μέσω της οποίας υπάρχει άμεση πρόσβαση σε όλα τα bitcoin της εφαρμογής.

- **read\_transactionfile** : Υλοποιείται η διαδικασία διαβάσματος του αρχείου transactionsFile που δίνεται ως όρισμα από την γραμμή εντολών. Ελέγχονται, τα ονόματα των χρηστών και εάν υπάρχει το απαιτούμενο ποσό από την πλευρά του αποστολέα. Επιπλέον, ελέγχεται η περίπτωση για διπλότυπα transaction\_ids καθώς και η ημερομηνία της κάθε συναλλαγής να είναι μεταγενέστερη της τελευταίας επιτυχώς εκτελεσμένης. Τέλος υλοποιείται ή απορρίπτεται η συναλλαγή. Η πολιτική που ακολουθείται για την επιλογή των bitcoins που θα σταλούν είναι του πρώτου διαθέσιμου και στην συνέχεια του επόμενου του για την συμπλήρωση του απαιτούμενου ποσού.
  - **request\_Transaction** : Ύστερα από ελέγχους υλοποιείται η συναλλαγή μεταξύ δυο χρηστών, όπως αυτή δόθηκε από την γραμμή εντολών.
  - **request\_Transactions** : Αφορά τις δύο συναρτήσεις για το διάβασμα μιας ακολουθίας συναλλαγών είτε από την γραμμή εντολών είτε από ένα δοσμένο αρχείο. Μετά τον απαραίτητο έλεγχο καλείται επαναληπτικά η request\_Transactions για ακολουθία συναλλαγών από την γραμμή εντολών ή αντίστοιχα η read\_requestTransactions\_from\_file, από την οποία καλείται επαναληπτικά η requestTransactions για τις συναλλαγές που περιγράφονται εντός του αρχείου.
  - **find\_Earnings\_Payments** : Υλοποιούνται αντίστοιχα οι 2 συναρτήσεις της εκφώνησης στις οποίες μέσω ελέγχου και μέσω των κλήσεων των συναρτήσεων find\_Transactions και check\_date και check\_time, οι οποίες επιτελούν λειτουργία αντίστοιχη του ονόματος τους.
  - **WalletStatus** : Υλοποιείται η αντίστοιχη λειτουργία της εκφώνησης και μέσω της calc\_wallet\_balance υπολογίζεται το ποσό που περιέχεται στο πορτοφόλι του χρήστη.
  - **BitcoinStatus** : Διατρέχεται κατά βάθος, DFS, το δέντρο του ιστορικού του bitcoin και υπολογίζονται πόσοι κόμβοι έχουν αριστερά παιδιά, δηλαδή συναλλαγές με άλλους χρήστες. Στη συνέχεια, ελέγχεται αν ο δεξιότερος κόμβος του bitcoin δεν έχει αριστερό παιδί, δηλαδή συναλλαγή και άρα είναι unspent.
  - **Tracecoin** : Αντίστοιχα με παραπάνω διατρέχεται το δέντρο του ιστορικού του bitcoin κατά βάθος και τυπώνονται οι αντίστοιχες συναλλαγές. Για να μην τυπωθούν και οι κόμβοι που αφορούν το υπόλοιπο ποσό που παραμένει στον αποστολέα, ρωτήστε με στην προφορική εξέταση να σας το εξηγήσω.
  - **read\_functions** : “Εν Αρχή ην το Χάος” κατά τον Ησίοδο, έτσι και στην συνάρτηση αυτή καλούνται όλες οι συναρτήσεις αρχικοποίησης του προγράμματος, στη συνέχεια οι συναρτήσεις που επιτελούν τις διάφορες λειτουργίες – απαιτήσεις της εργασίας και τέλος οι συναρτήσεις εξόδου και αποδέσμευσης της μνήμης.
- **Παρατηρήσεις και περιορισμοί της υλοποίησης της εργασίας :**
- **Makefile** : Χρησιμοποιείται Makefile και ικανοποιείται η απαίτηση για separate compilation.
  - **Διαχείριση μνήμης** : Κατά την εκτέλεση του προγράμματος με όλες τις παραμέτρους δεν διαπιστώθηκε κάποια απώλεια μνήμης ή κάποιο άλλου τύπου error. Οι παραπάνω έλεγχοι διενεργήθηκαν με τη χρήση της εντολής valgrind και των παραγώγων παραμέτρων της.
  - **Data duplication** : Σε μεγάλο βαθμό έχει αποφευχθεί παρουσιάζω το σχεδιασμό.
    - Κάθε sender και receiver συνδέονται στο ίδιο transaction και από εκεί προς το ίδιο κόμβο του δέντρου του ιστορικού των bitcoin που αφορούν τη συναλλαγή. Ο διαχωρισμός για το ρόλο του καθενός στη συναλλαγή επιτελείται μέσω ειδικών συναρτήσεων.
    - Κάθε bucket συνδέεται με ένα Wallet το οποίο δείχνει προς 2 Λίστες, Συναλλαγών και Bitcoin. Η Λίστα Bitcoin δείχνει προς ένα κόμβο του δέντρου συναλλαγών στον οποίο υπάρχει ένας δείκτης προς το όνομα του κατόχου και η αξία του bitcoin που κατέχει.

- Χρησιμοποιούνται 2 βοηθητικές Λίστες που αφορούν το σύνολο των έγκυρων συναλλαγών και το σύνολο των Bitcoin της εφαρμογής. Η χρησιμότητά τους είναι η άμεση πρόσβαση προς τα Bitcoin και τις συναλλαγές ώστε να γίνεται έλεγχος για διπλότυπα ή άλλες παρόμοιες λειτουργίες.
- Παρακάτω παρουσιάζεται ένα σχεδιάγραμμα όπου απεικονίζεται η εφαρμογή της εργασίας μου:

