ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
www.cslab.ece.ntua.gr

---

# ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ ΣΤΑ ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

Ακ. έτος 2020-2021, 9ο Εξάμηνο, Σχολή ΗΜ&ΜΥ

Τελική Ημερομηνία Παράδοσης: 2 εβδομάδες μετά το τέλος της εξεταστικής

## Εισαγωγή - Σκοπός

Το blockchain είναι η τεχνολογία πίσω από τα περισσότερα κρυπτονομίσματα και αποτελεί στην πραγματικότητα μια κατανεμημένη βάση που επιτρέπει στους χρήστες της να κάνουν δοσοληψίες (transactions) μεταξύ τους με ασφάλεια, χωρίς να χρειάζονται κάποια κεντρική αρχή (π.χ. τράπεζα).

Σκοπός της εργασίας είναι να φτάξετε το noobcash, ένα απλό σύστημα blockchain, όπου θα καταγράφονται οι δοσοληψίες μεταξύ των συμμετεχόντων και θα εξασφαλίζεται το consensus με χρήση Proof-of-Work.

Συνοπτικά, το σύστημά σας θα πρέπει να υλοποιεί τις παρακάτω λειτουργίες:

1. Ο κάθε χρήστης του noobcash (το κάθε process δηλαδή) θα έχει ένα noobcash wallet για να να πραγματοποιεί transactions. Το κάθε wallet αποτελείται από: (α) ένα private key γνωστό μόνο στον χρήστη-κάτοχο του wallet, που του επιτρέπει να ξοδέυει χρήματα (να στείλει δλδ χρήματα σε άλλο wallet) και (β) το αντίτοιχο public key, που απαιτείται για να λάβει ο κάτοχος του wallet χρήματα από άλλον χρήστη. Το public key είναι και η διεύθυνση του wallet του χρήστη.
2. Ο κάθε κάτοχος wallet μπορεί να κάνει συναλλαγές ξοδεύοντας NBC (noobcash coins). Αν η Αλίκη θέλει να στείλει 1 NBC στον Μπόμπ, τότε θα δημιουργήσει ένα transaction που θα περιέχει το ποσό που θέλει να στείλει και τη διεύθυνση του wallet του Μπομπ , δλδ το public key του Μπομπ. Το transaction αυτό θα το υπογράψει χρησιμοποιώντας το private key της.
3. Το κάθε transaction που δημιουργείται γίνεται broadcast σε όλο το δίκτυο noobcash.
4. Οι miners που λαμβάνουν το νέο transaction το επικυρώνουν: Στο παράδειγμά μας, πρώτα ελέγχουν ότι το transaction προήλθε από την Αλίκη χρησιμοποιώντας το public key της, πιστοποιούν ότι η Αλίκη έχει αρκετά χρήματα στο wallet της για να εκτελέσει το transaction (1 NBC στην περίπτωσή μας) και αν όλοι οι έλεγχοι επιτύχουν, προσθέτουν το transaction στο τρέχον block. Για την παρούσα άσκηση θεωρούμε ότι όλοι οι χρήστες είναι και miners.
5. Όταν το τρέχον block γεμίσει, τότε οι miners ξεκινούν τη διαδικασία mining με proof-of-work. Όποιος βρει το σωστό nonce κάνει broadcast το επικυρωμένο block σε όλο το δίκτυο, ώστε να γίνει γνωστό σε όλους τους nodes που συμμετέχουν.

6. Σε περίπτωση που 2 ή περισσότεροι miners κάνουν ταυτόχρονα mine ένα block, οι παραλήπτες των διαφορετικών αυτών blocks προσθέτουν στην αλυσίδα τους το πρώτο block που λαμβάνουν. Αυτό μπορεί να οδηγήσει σε 2 ή περισσότερες διακλαδώσεις της αλυσίδας. Για να καταλήξουν τελικά όλοι οι κόμβοι με την ίδια αλυσίδα του blockchain, τρέχουν τον αλγόριθμο consensus, σύμφωνα με τον οποίον σε περίπτωση conflict υιοθετούν την αλυσίδα με το μεγαλύτερο μέγεθος.

## Δίκτυο

Υποθέτουμε ότι το δίκτυό μας έχει μέγεθος n, δηλαδή n κόμβοι συμμετέχουν στο δίκτυο με ids από 0 έως n-1. Ο κάθε κόμβος του δικτύου μας θα πρέπει να μπορεί να ανταλλάσει μηνύματα με όλους τους υπόλοιπους κόμβους του συστήματος γνωρίζοντας την ip address και το port στο οποίο ακούνε. Για λόγους απλότητας, θεωρούμε ότι κάθε κόμβος διατηρεί λίστα με όλους τους υπόλοιπους κόμβους που συμμετέχουν κάθε στιγμή στο noobcash σύστημα. Η επικοινωνία μπορεί να υλοποιηθεί είτε με sockets (TCP ή UDP) είτε με REST api, χρησιμοποιώντας όποια βιβλιοθήκη θέλετε.

Ο πρώτος κόμβος του δικτύου, ο bootstrap node (id 0), είναι και ο κόμβος που δημιουργεί το genesis block, το πρώτο δλδ block του blockchain μας. Στο genesis block θέτουμε previous_hash=1, nonce=0 και η λίστα από transactions περιλαμβάνει μόνο ένα transaction που δίνει στον bootsrap κόμβο 100*n NBC coins από την wallet διεύθυνση 0. Αυτό είναι το μοναδικό block που δεν επαληθεύεται. Σταδιακά προστίθενται οι υπόλοιποι n-1 κόμβοι του δικτύου.

Για να εισαχθεί ένας νέος κόμβος στο σύστημα επικοινωνεί πρώτα με τον bootstrap κόμβο, του οποίου θεωρούμε γνωστά σε όλους τα στοιχεία επικοινωνίας ip address/port, και του στέλνει το public key του wallet του. Από τον bootstrap κόμβο λαμβάνει το μοναδικό id του (ο πρώτος το id 1, ο δεύτερος το id 2 κ.ο.κ) . Όταν εισαχθούν όλοι οι κόμβοι, ο bootstrap κάνει broadcast σε όλους τα ζεύγη ip address/port καθώς και τα public keys των wallets όλων των κόμβων που συμμετέχουν στο σύστημα. Θεωρείστε ότι από εδώ και πέρα δεν υπάρχουν εισαγωγές ή αποχωρήσεις κόμβων από το σύστημα. Επιπλέον, κάθε νέος κόμβος λαμβάνει από τον bootstrap κόμβο το blockchain όπως έχει διαμορφωθεί μέχρι εκείνη τη στιγμή, το κάνει validate και μετά από αυτό το σημείο μπορεί να κάνει transactions.

Ο bootstrap κόμβος, όπως αναφέρθηκε παραπάνω, έχει 100*n NBC coins από το genesis block. Μετά τη σύνδεση κάθε νέου κόμβου στο δίκτυο, ο bootstrap κόμβος εκτελεί ένα transaction όπου του μεταφέρει 100 NBC. Έτσι, μετά την εισαγωγή όλων των κόμβων, ο καθένας τους έχει 100 NBC στο wallet του.

## Noobcash backend

Στο noobcash backend υλοποιείται όλη η λογική του blockchain. Τα βασικά συστατικά στοιχεία του noobcash blockchain είναι τα παρακάτω:

**Block**

Το κάθε noobcash block έχει τις εξής πληροφορίες

- index: ο αύξων αριθμός του block,
- timestamp: το timestamp της δημιουργίας του block
- transactions:  Η λίστα με τα transactions που περιέχονται στο block
- nonce: η λύση του proof-of-work
- current_hash: το hash του block
- previous_hash: το hash του προηγούμενου block στο blockchain.

Θεωρούμε ότι το κάθε block έχει συγκεριμένη χωρητικότητα σε αριθμό από transaction. Η χωρητικότητα καθορίζεται από τη σταθερά capacity.

**Blockchain**

Η λίστα από blocks που έχουν επαληθευτεί.

**Wallet**

Ένα wallet συνδέεται με ένα ζεύγος public/private key. Το public key δρα ως διεύθυνση του wallet, την οποία μπορεί κανείς να μοιραστεί με οποιονδήποτε προκειμένου να μπορεί να δεχτεί πληρωμές. Το private key χρησιμοποιείται για να υπογράφονται τα transactions, ώστε να εξασφαλίζεται ότι μόνο ο κάτοχος του wallet (δλδ ο κάτοχος του αντίστοιχου private key) μπορεί να ξοδέψει χρήματα από το wallet. Αυτό γίνεται με τη συνάρτηση `sign_transaction()`. Οποιοσδήποτε γνωρίζει το public key ενός wallet μπορεί να επαληθεύσει ότι ένα transaction που ξοδεύει χρήματα από το wallet αυτό έχει δημιουργηθεί από τον κάτοχό του με την `verify_transaction()`.

**Transaction**

Κάθε transaction περιέχει πληροφορίες για μεταφορά χρημάτων από ένα wallet σε ένα άλλο. Οι πληροφορίες που περιλαμβάνει είναι

sender_address: Το public key του wallet από το οποίο προέρχονται τα χρήματα

receiver_address: Το public key του wallet στο οποίο θα καταλήξουν τα χρήματα

amount: το ποσό που θα μεταφερθεί

transaction_id: το hash του transaction

transaction_inputs: λίστα από Transaction Input (βλέπε παρακάτω)

transaction_outputs: λίστα από Transaction Output (βλέπε παρακάτω)

Signature: Υπογραφή που αποδεικνύει ότι ο κάτοχος του wallet δημιούργησε αυτό το transaction.

Ένα transaction μπορεί να δημιουργηθεί από τον κάτοχο του wallet από όπου θα μεταφερθούν τα χρήματα. Κάθε transaction γίνεται broadcast σε όλα τα μέλη του blockchain. Κατά τη λήψη ενός transaction από οποιονδήποτε node, καλείται η συνάρτηση `validate_transaction` που ελέγχει την ορθότητά του.

**Transaction Inputs/Transaction Outputs/Unspent Transactions**

Κάθε transaction περιέχει πληροφορία που δείχνει από ποιο/α προηγούμενο/α transaction προήλθαν τα χρήματα που μεταφέρονται. Αυτή η πληροφορία περιέχεται στο transaction input. Επίσης, κάθε transaction δημιουργεί outputs, που περιέχουν πληροφορία για το ποσό που έστειλε κάθε transaction στους εμπλεκόμενους. Στην περίπτωσή μας, θεωρούμε μόνο transactions που γίνονται ανάμεσα σε 2 wallets, επομένως κάθε transaction καταλήγει σε 2 transaction outputs: Ένα output για το receiver_address wallet με το ποσό του transaction κι ένα output για τον sender, με τα ρεστα, το ποσό δηλαδή που περίσσεψε. Μόνο unspent transaction outputs (UTXO), δηλαδή transaction outputs που δεν έχουν ξοδευτεί ήδη, μπορούν να χρησιμοποίηθούν ως input σε άλλο transaction προκειμένου να αποφευχθεί double spending.  Επομένως πάντα χρειάζονται UTXOs για να γίνει ένα transaction.

Σύμφωνα με τα παραπάνω, το Transaction Input αποτελείται απο το πεδία previousOutputId που είναι το id του Transaction Output από όπου προήλθε το ποσό που μεταφέρεται.

Το Transaction Output περιλαμβάνει: ένα μοναδικό αναγνωριστικό id, το id του transaction από το οποίο προέρχεται, τον recipient του transaction (τον νέο κάτοχο των coins) και το ποσό που μεταφέρθηκε.

Τελικά το υπόλοιπο του wallet ενός χρήστη είναι το άθροισμα όλων των unspent transaction outputs που έχουν ως recipient το συγκεκριμένο wallet.

Για να μη χρειάζεται να διατρέχουμε όλο το blockchain για να ελέγξουμε αν τα inputs ενός transaction είναι πράγματι unspent transaction ouputs προηγούμενων, κάθε node κρατάει επιπλέον το σύνολο των UTXOs για κάθε wallet.

## Βασικές συναρτήσεις

Οι βασικές λειτουργίες του συστήματος περιλαμβάνουν:

`generate_wallet()`
Δημιουργεί νέο wallet, δλδ ξεύγος public/private key χρησιμοποιώντας τον κρυπτογραφικό αλγόριθμο RSA.

`create_transaction()`
Δημιουργείται ένα νέο transaction που περιέχει όλα τα στοιχεία που απαιτούνται. Εδώ το πεδίο transaction_inputs γεμίζει με τα Transaction Input που περιέχουν τα ids των UTXOs που απαιτούνται για να συμπληρωθεί το ποσό που θέλουμενα ξοέψουμε.

`sign_transaction()`
Υπογράφεται το transaction με το private key του wallet.

`broadcast_transaction()`
Το transaction αποστέλλεται με broadcast σε όλους τους κόμβους

`verify_signature()`
Επαληθεύεται η υπογραφή του transaction αμέσως μετά τη λήψη του

`validate_transaction()`
Επαληθεύεται η ορθότητα του transaction που έχει ληφθεί. Η επαλήθευση περιλαμβάνει (α) την επαλήθευση της υπογραφής (verify_signature) και (β) τον έλεγχο των transaction inputs/outputs για να εξασφαλίσουμε ότι το wallet αποστολέας έχει το ποσό amount που μεταφέρει στον παραλήπτη. Για να επιτευχθεί το (β) ελέγχεται αν τα transaction inputs είναι πράγματι unspent transactions, αν είναι αφαιρούνται από τη λίστα των UTXO του κόμβου. Δημιουργούνται τα δύο Transaction Outputs και προστίθενται στη λίστα στη λίστα UTXO του node μας.

`wallet_balance()`
Μπορούμενα βρούμε το υπόλοιπο οποιουδήποτε wallet προσθέτοντας όλα τα UTXOs που έχουν παραλήπτη το συγκεκριμένο wallet.

`mine_block()`
Η συνάρτηση αυτή καλείται μόλις capacity transactions έχουν ληφθεί και επαληθευτεί από κάποιον κόμβο και υλοποιεί το proof of work δοκιμάζοντας διαφορετικές τιμές της μεταβλητής nonce και hashάροντας το block μέχρι το hash που θα προκύψει να αρχίζει από έναν συγκεκριμένο αριθμό από μηδενικά. Ο αριθμός αυτός καθορίζεται από τη σταθερά difficulty.

`broadcast_block()`
Μόλις βρεθεί ο κατάλληλος nonce, ο κόμβος κάνει broadcast το επαληθευμένο block σε όλους τους υπόλοιπους κόμβους.

`validate_block()`
Αυτή η συνάρτηση καλείται από τους nodes κατά τη λήψη ενός νέου block (εκτός του genesis block). Επαληθεύεται ότι (a) το πεδίο current_hash είναι πράγματι σωστό και ότι (b) το πεδίο previous_hash ισούται πράγματι με το hash του προηγούμενου block.

`validate_chain()`
Αυτή η συνάρτηση καλείται από τους νεοεισερχόμενους κόμβους, οι οποίοι επαληθεύουν την ορθότητα του blockchain που λαμβάνουν από τον bootstrap κόμβο. Στην πραγματικότητα καλείται η `validate_block` για όλα τα blocks εκτός του genesis.

`resolve_conflict()`
Αυτή η συνάρτηση καλείται όταν ένα κόμβος λάβει ένα block το οποίο δεν μπορεί να κάνει validate γιατί το πεδίο previous_hash δεν ισούται με το hash του προηγούμενου block. Αυτό μπορεί να σημαίνει ότι έχει δημιουργηθεί κάποια διακλάδωση, η οποία πρέπει να επιλυθεί. Ο κόμβος ρωτάει τους υπόλοιπους για το μήκος του blockchain και επιλέγει να υιοθετήσει αυτό με το μεγαλύτερο μήκος.

## Noobcash client

Θα πρέπει να υλοποιήσετε έναν client (ένα απλό cli αρκεί, αλλά αν θέλετε να είστε πιο fancy θα μετρήσει θετικά) που θα δίνει στον χρήστη τη δυνατότητα να εκτελεί τα παρακάτω:

*t <recipient_address> <amount>*

New transaction: Στείλε στο recipient_address wallet το ποσό amount από NBC coins που θα πάρει από το wallet sender_address. Θα καλεί συνάρτηση `create_transaction` στο backend που θα υλοποιεί την παραπάνω λειτουργία.

*view*

View last transactions: Τύπωσε τα transactions που περιέχονται στο τελευταίο επικυρωμένο block του noobcash blockchain. Καλεί τη συνάρτηση `view_transactions()` στο backend που υλοποιεί την παραπάνω λειτουργία.

*balance*

Show balance: Τύπωσε το υπόλοιπο του wallet.

*help*

Επεξήγηση των παραπάνω εντολών.

## Πειράματα

Θα αναπτύξετε το noobcash σε όποια γλώσσα προγραμματισμού θέλετε. Θα το στήσετε σε υποδομή του εργαστηρίου (θα σας δοθούν ακριβείς οδηγίες). Για την αναφορά θα εκτελέσετε τα παρακάτω πειράματα:

1) Απόδοση του συστήματος

   Θα στήσετε ένα noobcash με 5 clients. Αφού όλοι εισέλθουν στο σύστημα, ο καθένας θα διαβάσει το αρχείο transactionX.txt, όπου X το node id του κόμβου. Το κάθε αρχείο περιέχει transactions προς άλλους κόμβους με τη μορφή

   < recipient_node_id>  <amount>

   Έτσι ο κάθε κόμβος δημιουργεί και στέλνει στο δίκτυο ένα transaction ανά γραμμή. Η διαδικασία αυτή θα γίνει ταυτόχρονα για όλους τους clients. Για (a) capacity 1, 5 και 10 και (b) difficulty 4 και 5 θα καταγράψετε τα παρακάτω:

   - Throughput (ρυθμαπόδοση) του συστήματός σας, δηλαδή πόσα transactions εξυπηρετούνται στην μονάδα του χρόνου.
   - Block time, δλδ τον μέσο χρόνο που απαιτείται για να προστεθεί ένα νέο block στο blockchain.

2) Κλιμακωσιμότητα του συστήματος

   Θα επαναλάβετε το πείραμα για 10 clients και θα συγκρίνετε με τα προηγούμενα αποτελέσματα παρουσιάζοντας σε γράφημα τις μετρικές του προηγούμενου πειράματος (άξονας y) σε σχέση με τον αριθμό των clients (άξονας x).

Δουλέψτε σε ομάδες 2-3 ατόμων. Παραδοτέο της άσκησης θα είναι ο πηγαίος κώδικας (tarball με τα σχετικά αρχεία) καθώς και ένα ηλεκτρονικό κείμενο (pdf, docx ή odt) που θα παρουσιάζει τον

σχεδιασμό του συστήματός σας και τα αποτελέσματα των πειραμάτων. Ο κώδικας και η αναφορά θα παραδοθούν ηλεκτρονικά στην ιστοσελίδα του μαθήματος. Επίδειξη της άσκησης θα γίνει σε συνεννόηση με τους διδάσκοντες μετά τη λήξη της προθεσμίας για το παραδοτέο.