FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

LABORATORIJSKA VJEŽBA 4

Bitcoin Script i konsenzusni mehanizam

Sadržaj

1. Uvod

Bitcoin mreža je nastala s ciljem da se suprotstavi centraliziranim bankama, a danas je već svima poznata činjenica. Svi podaci o transakcijama pohranjuju se u blockchainu, koji se sastoji od niza ulančanih blokova, svaki zaštićen kriptografskim kodom. Uz to, svaki sudionik mreže je anoniman te ima pohranjene sve podatke o transakcijama u svojoj memoriji.

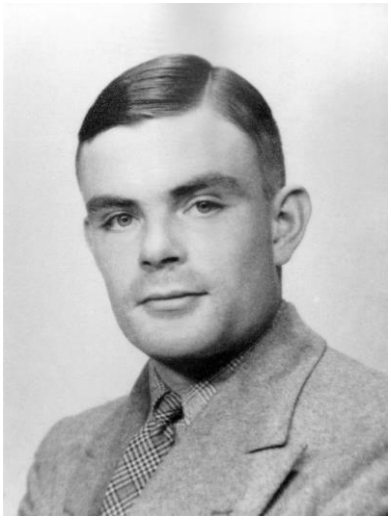
No, kako bi se postiglo sve navedeno, bilo je nužno uvesti dosad nezamislive algoritme, mehanizme, pravila i procedure. Bitcoin mrežu čine sudionici koji ne vjeruju bankama, drugim centraliziranim sustavima, pa čak ni sebi međusobno. Međutim, iz takvog nepovjerenja nastala je mreža u kojoj nevjerujući nikome, svi sudionici mogu vjerovati jedni drugima.

Bitcoin je uveo niz noviteta kao protutežu dominantnom centraliziranom monetarnom sustavu. Stoga, ova laboratorijska vježba će nam dati uvid u značajne inovacije koje je Bitcoin donio.

2. Bitcoin Script

Bitcoin Script je jednostavan programski jezik koji se koristi u Bitcoin transakcijama kako bi se definirali uvjeti pod kojima će se ta transakcija izvršiti. Bitcoin Script je zapravo skup instrukcija koje se izvršavaju kako bi se provjerili uvjeti koji su definirani u skripti transakcije.

Korištenjem Bitcoin Scripta, korisnici mogu stvoriti složene transakcije koje se izvršavaju samo ako su zadovoljeni određeni uvjeti. Na primjer, korisnik može stvoriti transakciju koja se izvršava samo ako drugi korisnik pristane potpisati transakciju. Također je moguće stvoriti transakcije koje se izvršavaju samo ako se zadovolje određeni uvjeti u vremenskom periodu ili ako se koristi određeni ključ. Ovaj programski jezik namjerno spada u skupinu nekompletnih Turingovih jezika. Dizajniran je tako kako bi se izbjegle beskonačne petlje i trošilo previše mrežnih resursa. Alan Turing je prikazan na slici 1 a Turingova kompletnost znači da sustav može odraditi bilo kakav zadatak ukoliko mu se da dovoljno resursa i vremena.

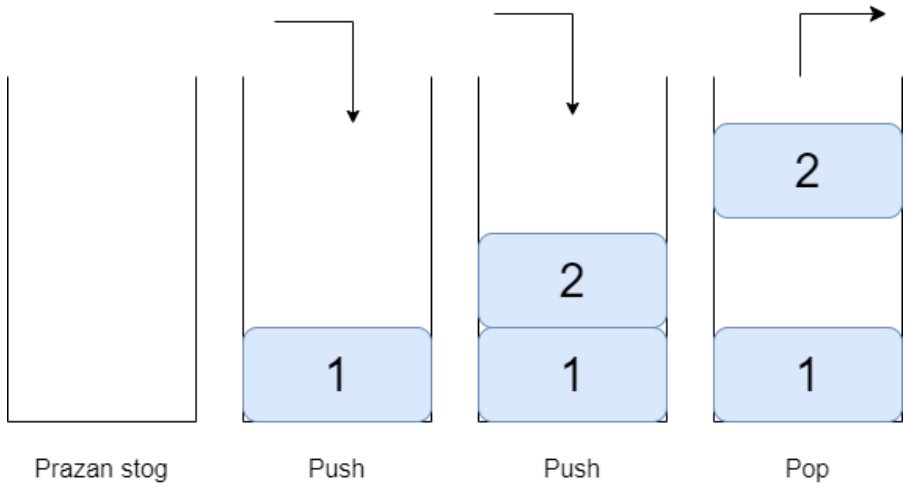


Slika 1. Alan Turing

Bitcoin Script se koristi kako bi se povećala sigurnost i fleksibilnost Bitcoin transakcija, a omogućava i stvaranje pametnih ugovora (smart contracts) unutar Bitcoin mreže. Međutim, zbog svoje složenosti i ograničenja u pogledu podržanih operacija, Bitcoin Script se obično koristi samo za jednostavne transakcije, dok se za složenije operacije koriste pametni ugovori na drugim blockchain platformama.

Važno je napomenuti da Bitcoin Script nije potpuni programski jezik, već jezik skriptiranja koji se koristi za izvršavanje specifičnih operacija u Bitcoin transakcijama. Stoga, neke od uobičajenih značajki koje se nalaze u drugim programskim jezicima, poput podrške za funkcije i biblioteke, možda neće biti dostupne u Bitcoin Script-u.

Bitcoin Script jezik ima jednostavnu sintaksu koja se sastoji od niza operacijskih kodova (eng. opcodes) koji se izvršavaju na stogu (eng. stack). Skica stoga i primjer rada prikazani su na slici 2.



Slika 2. Stog

Osnovni opcodovi u Bitcoin Scriptu su:

OP_0 - Ovo predstavlja nulu, tj. prazan stog.

OP_PUSHDATA1 - Ovaj op kod se koristi za dodavanje podataka na stog. Nakon ovog op koda, slijedi jedan bajt koji označava broj bajtova podataka koje treba dodati na stog.

OP_DUP - Duplicira vrh stoga.

OP_EQUAL - Uspoređuje dvije stavke s vrha stoga i vraća TRUE ako su iste, a FALSE ako nisu.

OP_EQUALVERIFY - Isto kao i OP_EQUAL, ali uklanja vrhove stoga ako nisu isti.

OP_HASH160 - Uzima vrh stoga, računa njegov SHA-256 hash i zatim njegov RIPEMD-160 hash, te umetne rezultat na stog.

OP_CHECKSIG - Uzima javni ključ i digitalni potpis s vrha stoga, te provjerava jesu li odgovarajući za transakciju. Ako jesu, vraća 1, a ako nisu, vraća 0.

OP_CHECKMULTISIG - Uzima više javnih ključeva i digitalnih potpisa s vrha stoga, te provjerava jesu li odgovarajući za transakciju. Ako jesu, vraća 1, a ako nisu, vraća 0.

Ovo su samo neki od osnovnih op kodova koji se koriste u Bitcoin Scriptu. Postoji puno drugih op kodova koji se koriste za razne druge svrhe, ali ovi su najčešće korišteni.

Bitcoin Script kod može se testirati u Visual Studio Code (VS Code) uz korištenje odgovarajućih dodataka (eng. extensions).

Postoje mnogi dodaci za VS Code koji podržavaju Bitcoin Script, poput Bitcoin Language Service (BLS) i Bitcoin VSCode Extension. Ovi dodaci pružaju različite značajke, uključujući sintaksnu provjeru, naglašavanje koda, automatsko dovršavanje koda, pomoć pri debugiranju i druge.

Da biste koristili ove dodatke, prvo ih morate instalirati iz trgovine dodataka (eng. extensions marketplace) u VS Code-u. Nakon instalacije, možete otvoriti Bitcoin Script datoteku i dodatak će automatski prepoznati jezik i pružiti odgovarajuće značajke.

Bitcoin Script datoteke najčešće nemaju specifičnu ekstenziju jer se skripte pohranjuju u transakcijama u binarnom obliku. Međutim, u nekim slučajevima može se koristiti ".script" ekstenzija datoteke kako bi se jasno označilo da se radi o Bitcoin Script datoteci.

3. Pay-to skripte

Pay-to skripte u Bitcoin-u su način slanja bitcoina koji koristi Bitcoin Script, koji se koristi za definiranje uvjeta potrebnih za potrošnju bitcoina.

Kod Pay-to skripti, adresama za primanje bitcoina dodaju se uvjeti koji moraju biti zadovoljeni da bi se mogli potrošiti bitcoini sa te adrese. Ovi uvjeti se zapisuju kao skripte koje se provjeravaju pri potrošnji.

Korištenje Pay-to skripti pruža veću fleksibilnost u postavljanju uvjeta za potrošnju bitcoina te omogućava implementaciju složenijih scenarija potrošnje, kao što su transakcije koje zahtijevaju potpis više korisnika ili transakcije koje se mogu potrošiti samo u određenom vremenskom okviru.

P2PK, P2PKH i P2SH su tri osnovne vrste Pay-to-Script (P2SH) skripti u Bitcoinu.

P2PK (Pay-to-Public-Key) - U P2PK skripti, primaocu se šalje Bitcoin izravno na njegov javni ključ. Ovo se vrlo rijetko koristi jer nije sigurno zbog činjenice da je javni ključ lako ukrasti.

P2PKH (Pay-to-Public-Key-Hash) - Ovo je najčešća vrsta P2SH skripte u Bitcoinu. Umjesto slanja Bitcoin-a izravno na javni ključ, P2PKH skripta prima Bitcoin na adrese koje su povezane s javnim ključevima. Adrese se sastoje od hash vrijednosti javnog ključa, što je sigurnije nego slanje Bitcoin-a izravno na javni ključ.

P2SH (Pay-to-Script-Hash) - P2SH skripte su fleksibilne skripte koje koriste kompleksnije uvjete prije nego što se mogu otključati. To omogućuje korisnicima da stvore sigurnije i složenije uvjete plaćanja. Umjesto slanja Bitcoin-a na adresu koja je povezana s javnim ključem, P2SH skripta prima Bitcoin na adresu koja je povezana s hash vrijednošću skripte. To znači da se uvjeti plaćanja skripte ne otkrivaju dok se ne otključaju Bitcoin-i.

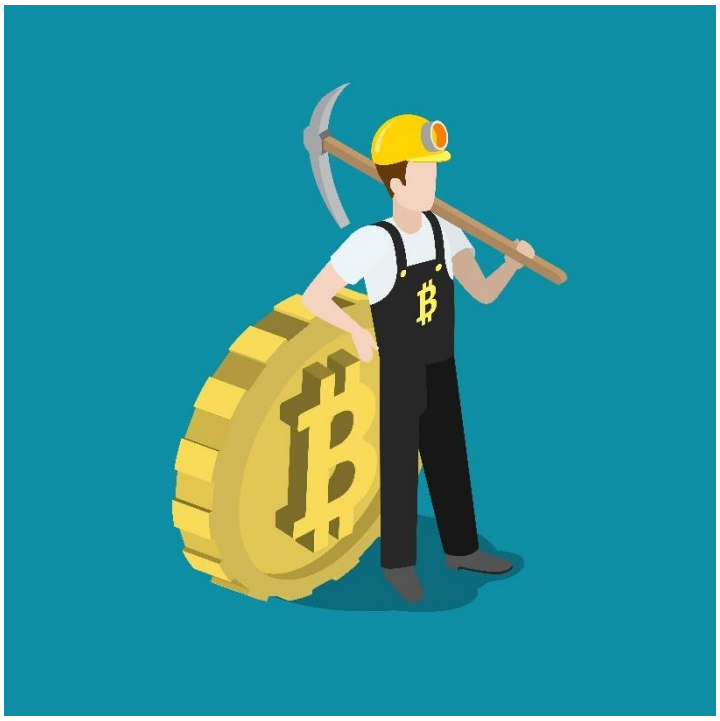
4. Proof-of-Work (PoW) mehanizam konsenzusa

Konsenzusni mehanizam je način na koji se postiže dogovor između članova mreže u distribuiranom sustavu poput blockchaina. Umjesto centralne vlasti koja bi kontrolirala sve transakcije, svi članovi mreže (čvorovi) moraju se složiti o tome koje su transakcije valjane i koje bi trebale biti dodane u blockchain.

Postoje različiti konsenzusni mehanizmi koji se koriste u blockchainima i drugim distribuiranim sustavima. Među popularnijim su Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), i Practical Byzantine Fault Tolerance (PBFT). Svi ovi mehanizmi imaju za cilj osigurati da se transakcije mogu sigurno izvršiti u distribuiranom okruženju uz minimalne troškove i bez potrebe za centralnom autoritetom.

Bitno je napomenuti da konsenzusni mehanizmi ne samo da omogućuju usklađivanje oko valjanih transakcija, već i sprečavaju manipulaciju mrežom. To znači da svaki član mreže može provjeriti valjanost transakcije i osigurati da se ne pokušava manipulirati ili krivotvoriti podaci. Kako bi se postigla sigurnost i pouzdanost distribuiranog sustava, bitno je odabrati pravi konsenzusni mehanizam koji će osigurati potrebnu razinu sigurnosti, skalabilnosti i učinkovitosti.

Mining (rudarenje) u Bitcoin-u je proces pronalaženja novih blokova koji se dodaju u blockchain. To se postiže rješavanjem kompleksnih matematičkih problema korištenjem računalne snage. Međutim, kako bi se spriječilo previše brzo dodavanje novih blokova i preplavljivanje mreže, Bitcoin koristi Proof of Work (PoW) mehanizam. Ilustracija rudara prikazana je na slici 3.



Slika 3. Rudar bitcoina

Proof of Work je konsenzusni mehanizam koji se koristi za potvrđivanje transakcija i stvaranje novih blokova u blockchainu. U PoW sustavu, rudari moraju riješiti matematički problem koji je teško riješiti, ali lako provjeriti rješenje. Kada se rješenje pronađe, rudar ga šalje u mrežu, a ostali članovi mreže provjeravaju rješenje. Ako je rješenje točno, blok se dodaje u blockchain, a rudar koji je pronašao rješenje dobiva nagradu u obliku novih bitcoina.

Dakle, Proof of Work je bitan za Bitcoin mining jer omogućuje mreži da se sama regulira, a rudari koji koriste računalnu snagu za rješavanje matematičkih problema dobivaju nagradu za svoj trud. U tom smislu, PoW mehanizam osigurava sigurnost i pouzdanost mreže, jer bi bilo vrlo teško manipulirati transakcijama bez obavljanja ovog teškog matematičkog zadatka.

U Bitcoin miningu, nonce je 32-bitni broj koji se dodaje u blok header kako bi se promijenio hash bloka i stvorio hash koji zadovoljava ciljanu težinu (target difficulty). Kada se blok kreira, rudar (miner) dodaje transakcije u blok, a zatim kreira blok header koji sadrži informacije o bloku, kao što su prethodni blok hash, transakcijski hash, vremenska oznaka i drugo. Nakon toga, miner mora dodati nonce vrijednost u blok header i stvoriti hash za cijeli blok. Ako je hash manji od ciljane težine, blok se smatra riješenim, i tada se mineru dodjeljuje nagrada u obliku novih Bitcoina.

Kako bi pronašli ispravan nonce, miner započinje s brojem 0 i inkrementira nonce za 1 sve dok ne pronađe ispravan hash za blok koji zadovoljava ciljanu teškoću. Ovo je izuzetno težak proces i uključuje isprobavanje ogromnog broja različitih nonce vrijednosti. Svaki put kada miner promijeni nonce vrijednost, stvara se potpuno novi hash bloka, što znači da je promjena noncea jedini način za mijenjanje blokova kako bi se povećale šanse za pronalazak ispravnog hasha. Stoga, nonce i hash su međusobno povezani u Bitcoin

miningu jer miner mora mijenjati nonce da bi stvorio hash koji zadovoljava ciljanu teškoću i riješio blok.

Trenutni Bitcoin težinski faktor (difficulty level) uzima u obzir ciljnu vrijednost koju hash vrijednost bloka mora zadovoljiti. Bitcoin mreža zahtijeva da hash vrijednost bloka počinje određenim brojem nula kako bi bio prihvaćen kao rješenje za Proof-of-Work algoritam.

Broj početnih nula koji se traži u hash vrijednosti bloka varira s težinom problema (difficulty level) i mijenja se svakih 2016 blokova. Trenutno (u travnju 2023.) ta vrijednost iznosi otprilike 18 nula.

5. ZADACI

1. Pokrenuti proces rudarenja (mining procces) u PoW sustavu. Sustav mora imati prilagodljiv difficulty level. Pokrenuti proces rudarenja na minimalno 4 difficulty levela (preporuka 4,5,6,7) po minimalno 2 puta. Izmjeriti vrijeme potrebno za izvođenje procesa. Dobivena vremena unijeti u tablicu, izračunati prosjek za svaki difficulty level i grafički prikazati. Potrebno je napisati kratak izvještaj u kojem se opisuje zadatak, teorija i korištena programska rješenja te kreirati tablicu i graf i napisati zaključak. Uz dobivene rezultate, ocjenjuje se i forma izvještaja.