

Cryptocurrency

În ecosistemul unei noi monede virtuale există mai multe entități ce întrețin dinamica rețelei ce implică crearea de tranzacții, validarea tranzacțiilor și în final salvarea acestora. Astfel:

- **Utilizatorii** monedei virtuale **crează tranzacții** ce implică un transfer din *wallet*-ul personal, către un alt *wallet*, și le adaugă într-o coadă pentru a fi procesate.
- **Supervizorii** preiau din coada propuneri de tranzacții și fac o serie de validări: există fonduri suficiente pentru a crea tranzacția (fonduri ale utilizatorului)? *wallet*-ul destinatar este valid? după care introduc tranzacția într-o nouă coadă pentru a fi adăugată în blockchain (istoricul tranzacțiilor)
- **Minerii** preiau tranzacțiile verificate de către supervizori și **le adaugă** într-un istoric (ordonat cronologic) al tranzacțiilor.

Scrieți un program prin care simulați acțiunile actorilor implicați știind că aceștia acționează concurent.

Cozile implicate în operații sunt limitate la o capacitate **X**, astfel încât dacă limita este atinsă nu se pot adăuga noi elemente până când nu se eliberează un loc.

O tranzacție propusă conține: **cod wallet utilizator (id thread)**, **valoare tranzacție**, **cod wallet destinatar (id thread)**

Procesarea de către supervizor a unei tranzacții propuse presupune validarea menționată mai sus:

- Pentru a valida că utilizatorul are destui bani pentru a onora valoarea din tranzacție, supervizorul trebuie să parcurgă tot istoricul tranzacțiilor care este stocat într-un fișier cu o structură ce se va detalia mai jos. Dacă suma tuturor tranzacțiilor istorice asociate *wallet*-ului utilizatorului este o valoare mai mare decât valoarea tranzacției propuse, atunci tranzacția este acceptată și se adaugă în următoarea coadă pentru a fi procesată de mineri. În cazul în care valoarea este mai mică, tranzacția este respinsă.
- Pentru a valida dacă codul *wallet*-ului destinatar este corect, se va face o verificare bazată pe numărul total de utilizatori definiți în program, acest număr oferind ca și informație lista tuturor codurilor disponibile (30 utilizatori, id-urile sunt [0, 29])

Tranzacția acceptată conține: **cod wallet utilizator (id thread)**, **valoare tranzacție**, **cod wallet destinatar (id thread)**, **id supervizor**

Minerul preia din coada tranzacțiile acceptate și simulează procesarea tranzacției prin scrierea informațiilor primite într-un fișier denumit **blockchain.txt**. Acest fișier va fi folosit de către **Supervizori** pentru a face validarea tranzacției prin parcurgerea întregului istoric și agregând toate tranzacțiile asociate *wallet*-ului asociat utilizatorului ce a inițiat tranzacția.

Acțiunile utilizatorilor, supervizorilor, minerilor se afișează pe ecran:

“Tranzacția inițiată de wallet X a fost preluată de supervizorul Y”

“Tranzacția inițiată de wallet X cu valoarea Y către wallet Z a fost acceptată/respinsă de supervizorul T, sold total înainte de operație: W”

“Minerul X a salvat tranzacția inițiată de wallet X cu valoarea Y către wallet Z, validată de supervizorul T”

Programul se termină când toate tranzacțiile inițializate sunt acceptate/refuzate și adăugate în **blockchain.txt**.

Considerăm

- N utilizatori
- M supervizori
- 1 miner

- Dimensiunea cozilor: 20
- Fiecare wallet incepe cu 100 unitati (se pot adauga cate o tranzactie in blockchain.txt pentru fiecare utilizator pentru a simula acest lucru)
 - 1,100,1,-1
 - 1,100,2,-1
 -
 - 1,100,N,-1

Caz de testare:

- N=50
- M=4
- Utilizatorii adaugă tranzacții (fiecare utilizator o singura tranzactie, dupa care isi termina executia) - tranzacțiile se genereaza cu valori random (cod wallet utilizator este id-ul thread-ului asociat utilizatorului, valoare tranzacții: random **1-200**, cod wallet destinatar: random 0-70).
- Dacă cozile sunt pline, fie utilizatorii, fie supervizorii așteaptă până când sunt notificati ca exista un loc disponibil..
- Supervizorii preiau din coada tranzacții, verifica dacă exista fonduri suficiente si transmit tranzacția către procesarea in blockchain prin adaugarea ei coada corespunzătoare. La sfarsit notifică faptul ca tranzacția a fost procesata si există un loc disponibil in coada destinata tranzacțiilor create de utilizatori.
- Minerul adauga tranzactia acceptata in **blockchain.txt** si notifică faptul ca exista un loc disponibil in coada dedicată tranzacțiilor acceptate.

Observatie: pentru așteptare condiționată este necesar sa se folosească un mecanism de tip “wait-notify” (busy-waiting nu este acceptabil).