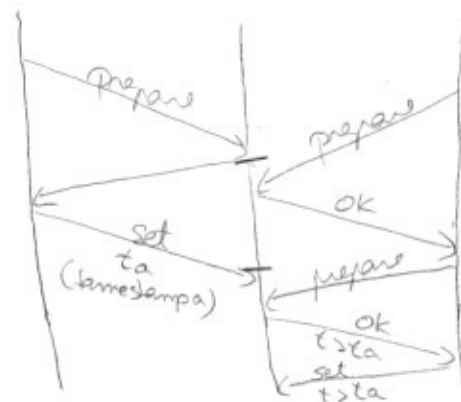
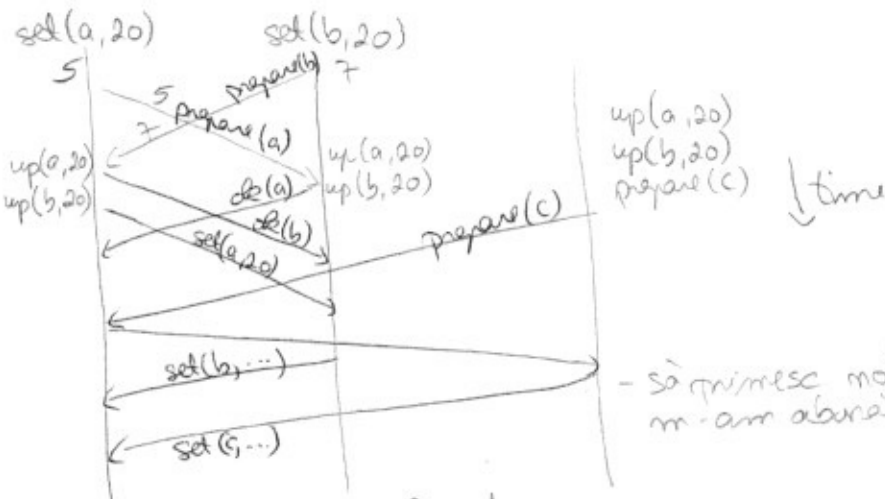


SEMINAR 6

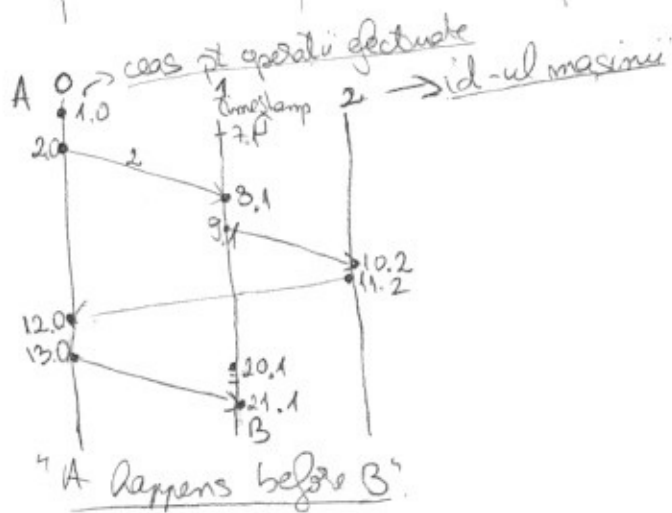
Problema lab 10

functia a=10 functia b=10
 set(a, 20) ^{timestamp} set(b, 20) ⁷
~~a=20~~ ~~b=10~~
 updated(a, 20) updated(a, 20)
 updated(b, 20) updated(b, 20)

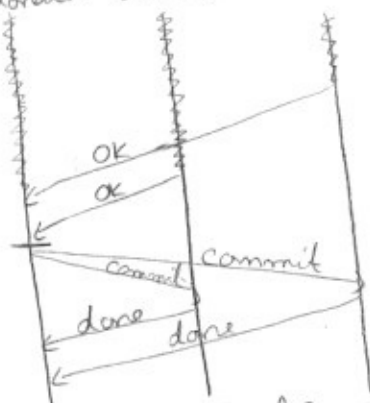
set(c, 12)
 up(c, 12)
 set(c, 16)
 up(c, 16)



- să primesc modificări doar de la variabilele la care m-am abonat.



Tranzacții distribuite: → un coordonator cu mai multe procese
 coordonator subordonat subordonat



SCHEMĂ ÎN 2 PAȘI

- coordonatorul centralizează mesajele: dacă toate mesajele vin cu OK și niciunul cu FAIL, se transmite COMMIT, altfel în momentul în care vin toate cu mesajul DONE, atunci se încheie tranzacția
 - nu se face rollback din propria inițiativă, decât dacă coordonatorul cere asta
 - coordonatorul poate genera un rollback zicând înainte de commit.

EXCEPTION SAFETY.

- 3 niveaux de exception safety

← basic guarantee = lesà sistemul într-o stare validă (no resources are leaked)
← strong guarantee = se face rollback la orice modificare { if commit (no fail)
if abort (no fail)
← no throw (no fail) = destructori

MPI - multiplication of 2 polynomials karatsuba → mergeSort example.