

1. ariketa [1,5 puntu]

Sistema banatu batean S zerbitzari bat dago UTC denbora fidagarria eskaintzen duena. Beste nodo guztiek bere erloju lokala periodikoki sinkronizatzen dute S zerbitzariarekiko, Cristianen algoritmoa erabiliz. Horretarako, `get_time()` sistema-deia erabiltzen dute. Dei honek denbora lokala bueltatzen du, milisegundotako bereizmenarekin.

Bedi ondorengo exekuzioa, C bezero nodo baten sinkronizazioa erakusten duena:

```
...
T0= get_time();
send(S, eskaera);
receive(S, &tm);
T1= get_time();
/* C nodoko erloju lokalaren sinkronizazioa */
```

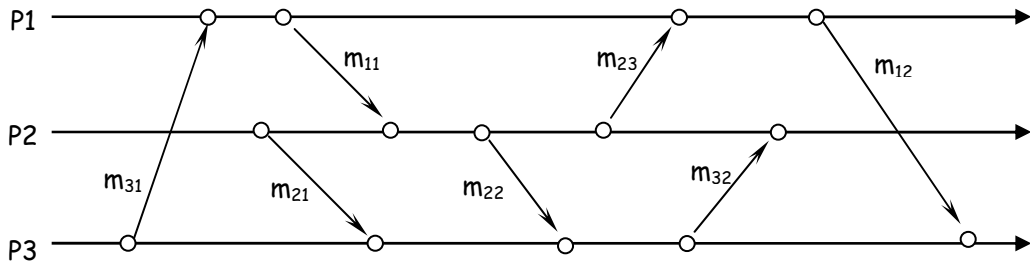
Bedi $T_0=2:30:59.370$ eta $T_1=2:30:59.390$. S zerbitzariak bueltatutako denbora tm -n $2:30:59:320$ da.

- (a) Kalkula ezazu C nodoak lortutako prezisioa sinkronizazio honetan. Arrazoitu erantzuna.
- (b) Arrazoitu ezazu ea NTPko algoritmo simetrikoa erabili ezkerro orokorrean prezisio hobea lor daitekeen.
- (c) Kalkula ezazu C nodoko erlojuaren desbiderapena S zerbitzariarekiko. Arrazoitu erantzuna.

- (d) Sinkronizazioa gauzatzeko, erloju lokalaren abiadura `adjust_clock(adjust_direction, interval)` sistema-deiaren bitartez aldatu daiteke, non `adjust_direction` abiadura-aldaketa adierazten duen balio boolearra da (AZKARTU/MANTSOTU). Dei honek zera egiten du: `interval` segundotan erloju lokala azkartzen edota mantsotzen du %0,1ean, ondoren bere abiadura normala berreskuratuz. Eman ezazu balio egokiak `adjust_clock()` deiaren argumentuentzako, aurreko ataleko datuen arabera C nodoak sinkronizazioa burutu dezan. Arrazoitu ezazu ea denboraren monotonizitate propietatea bermatzen den.
- (e) Nodoen erlojuen deriba maximoa 10^{-5} ekoa dela ezagutzen da. Sistema honetan exekutatzen diren aplikazioek gehienez 100 ms-tako desbiderapenak onartzen dutela ere ezagutzen da. Hau honela izanik, eta aurreko ataleko datuen arabera sinkronizazioa gauzatu ondoren, zein unerako planifikatu beharko du C nodoak beranduenez bere hurrengo sinkronizazioa? Balioa emateaz aparte, azaldu ezazu grafikoki.

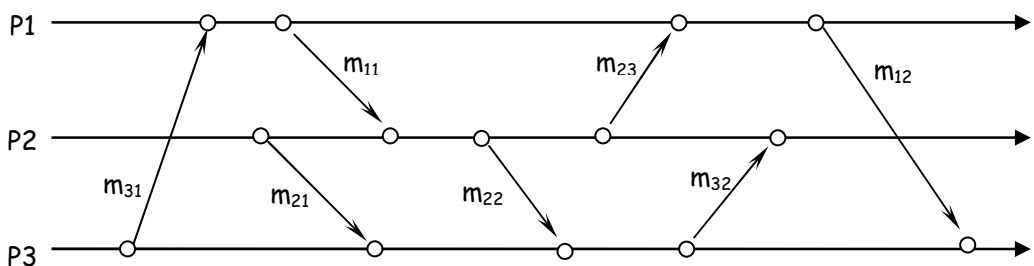
2. ariketa [2 puntu]

Hurrengo kronograman sistema banatu bateko hiru prozesuen exekuzioa agertzen da



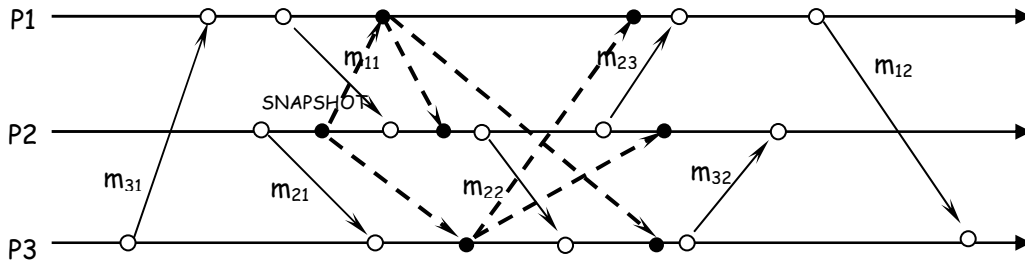
- (a) Jar ezazu kronograman Lamporten algoritmoa erabiliz gertaera bakoitzari dagokion denbora logikoa (hasieran denbora guztiak 0 balio dute).
- (b) Eman bi gertaera konkurrente non Lamporten denbora logikoez ahalbidetzen duten esatea konkurrenteak direla, eta beste bi gertaera konkurrente non Lamporten denbora logikoez EZ duten ahalbidetzen esatea konkurrenteak diren ala ez. Arrazoitu erantzuna.

- (c) Jar ezazu kronograman oraingoan denborazko bektoreen algoritmoa erabiliz gertaera bakoitzari dagokion denborazko bektorea (hasieran denbora guztiak 0 balio dute).



- (d) Denborako bektoreetan oinarrituz, arrazoitu ezazu (b) ataleko gertaerak konkurrenteak direla.

Hurrengo kronograman Chandy-Lamport algoritmoaren exekuzio bat agertzen da, sistemako egoera global sendoa kalkulatzeko. P2 prozesuak algoritmoari hasiera ematen dio, beste prozesuei SNAPSHOT mezua bidaliz (ez jarraian marraztutako geziak).



(e) Marraztu kronograman kalkulaturako egoera globalari dagokion moztea. Azaldu egoera globala zein den, prozesu bakoitzaren egoera adieraziz, baita komunikazio kanal bakoitzaren egoera ere.

(f) Denborazko bektoreetan oinarrituz, arrazoitu lortutako moztea sendoa dela.

(g) Marraztu kronograman sendoa ez den mozte bat, eta arrazoitu zergatik ez den sendoa.

3. ariketa [0,75 puntu]

(a) Elkarrekiko eskusio banatua eskaintzen duen Ricart eta Agrawalaren algoritmoa azaldu ezazu, N prozesu multzo batentzako. Algoritmo honen ezaugarrietako bat eskaera mezuei eskatzailearen denbora marka logikoa esleitzen zaiela da.

(b) Arrazoitu ezazu ea algoritmo honen hutsegite tolerantzia maila, elkarrekiko eskusiorako algoritmo zentralizatuarena (zerbitzari bakarrean oinarritutakoa) baino handiagoa edo txikiagoa den. Arrazoitu erantzuna.

4. ariketa [0,75 puntu]

Hedapen semantika desberdinen testuinguruan (talde-komunikazioa), arrazoitu ezazu ea ondorengo esaldi bakoitza egia ala gezurra den. Nahi baduzu, adibide baten bidez azaldu ezazu erantzuna:

(a) Hedapen guztiz ordenatuak, FIFO hedapena inplikutzen du.

(b) Hedapen kausalak, FIFO hedapena inplikutzen du.

(c) Hedapen guztiz ordenatuak, hedapen kausala inplikutzen du.

5. ariketa [1,5 puntu]

Datu-base zerbitzu batek honako espezifikazioa duten operazioak eskaintzen ditu:

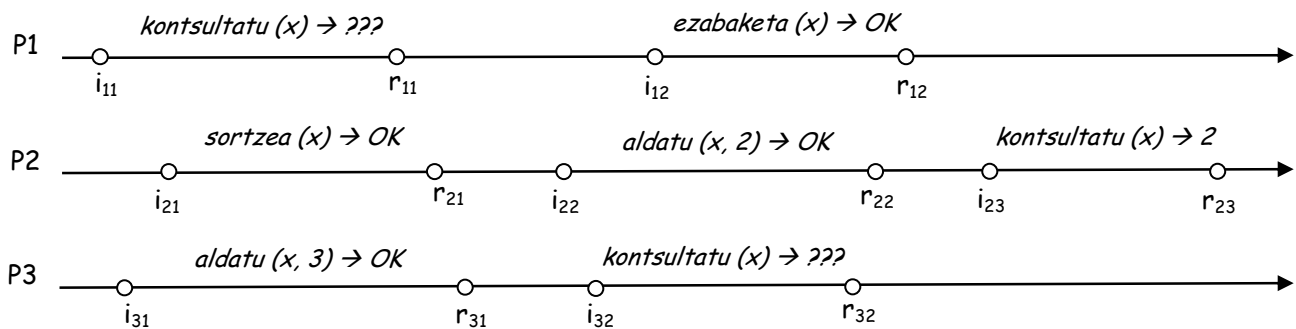
sortzea (x): if (ez da existitzen(x)) {sortu x ; $x = 0$; return OK} else return ERROREA;

aldatu (x, v): if (existitzen da(x)) { $x = v$; return OK} else return ERROREA;

kontsultatu (x): if (existitzen da(x)) return x ; else return ERROREA;

ezabaketa (x): if (existitzen da(x)) {ezabatu x ; return OK} else return ERROREA;

Bedi honako kronograma, hiru bezero prozesuen exekuzioa erakusten duena:



- (a) P1 prozesuaren kontsultarentzat (o_{11} operazioa), eman ezazu exekuzio legalei dagozkien erantzun balio posible guztiak. Gauza bera egizu P3 prozesuaren kontsultarentzat (o_{32} operazioa). Legala izango ez litzatekeen exekuzio bati dagozkion erantzun balio posibleak ere eman itzazu.
- (b) Eman ezazu sendotasun sekuentziala duen exekuzio legal bat. Arrazoitu erantzuna. Existituko ez balitz, arrazoitu erantzuna ere.
- (c) Eman ezazu lerrokagarria den exekuzio legal bat. Arrazoitu erantzuna. Existituko ez balitz, arrazoitu erantzuna ere.

6. ariketa [0,5 puntu]

Fitxategi sistema banatu batean, FS_i zerbitzari batek esportatzen duen karpeta bat muntatzerakoan hurrengo alternatibak kontsidera itzazu:

- A sistema: esportatutako karpeta `/mount/remote` izeneko karpeta lokalean muntatzea inposatzen du.
- B sistema: esportatutako karpeta `/mount/Ck` izeneko karpeta lokalean muntatzea inposatzen du, C_k karpeta muntatzen duen bezero makinaren izena delarik.
- C sistema: esportatutako karpeta `/mount/FSi` izeneko karpeta lokalean muntatzea inposatzen du, FS_i karpeta esportatzen duen zerbitzari makinaren izena delarik.
- D sistema: esportatutako karpeta bezero makinako fitxategi sisteman nahi den tokian muntatzen uzten du.

Aurkeztutako sistema mota bakoitzarentzat, arrazoitu ezazu ea bezero makinetan exekutatzen diren aplikazioei gardentasuna fitxategien identifikazioan eskaintzen zaien ala ez.

7. ariketa [1 puntu]

Hurrengo kodea:

```
main{
    FILE *fd;

    fd = fopen("fitx", "w"); // fitx fitxategia idazketa moduan ireki
    ...
    fwrite(fd, IP_ADDRESS); // fitxategian IP helbide lokala idatzi
    ...
    fclose(fd); // fitxategia itxi
    ...
    fd = fopen("fitx", "r"); // fitx fitxategia irakurketa moduan ireki
    ...
    fread(fd, &ip_address); // fitxategitik IP helbidea irakurri
    ...
    fclose(fd); // fitxategia itxi
}
```

sistema banatu batean era konkurrentean P1 eta P2 prozesuengatik exekutatua da, `fitx` fitxategia FS fitxategi zerbitzari batean kokatuta dagoelarik. P1 eta P2 prozesuak bezero makina desberdinetatik (IP helbide desberdina dutenak) exekutatzen badira, bi makinak FStik desberdinak direlarik,

- (a) Arrazoitu ezazu nola *write-through* politika huts batek, bezeroen cachearen sendotasuna kudeatzeko, ez den nahikoa bere baitan (hau da, baliotasun mekanismorik gabe) UNIX semantika eskaintzeko (edota UNIX semantikari hurbiltzeko).

- (b) Arrazoitu ezazu nola *write-on-close* politika huts batek (fitxategia zerbitzarian idazten da soilik bezeroan itxi egiten denean) ere ez den nahikoa sesio semantika eskaintzeko.

- (c) Aurreko kasu bietan, azaldu ezazu zein mekanismo gehigarri beharko liratekeen aipaturako semantikak eskaintzea lortzeko.