

Izena:

# Sistema Banatuak

2009ko ekaina

## 1. ariketa [1 puntu]

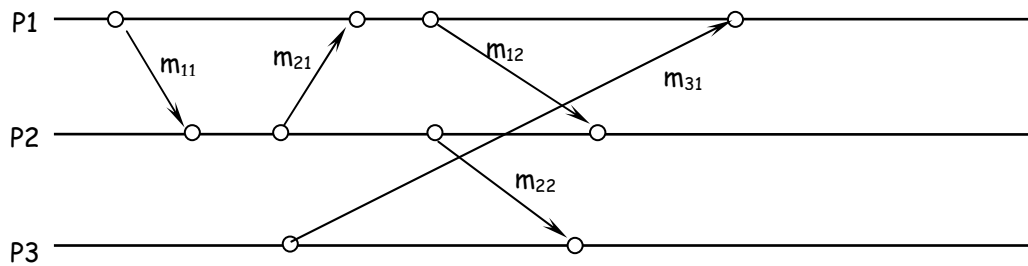
Sistema banatu baten erlojuen sinkronizazioa Cristianen algoritmoa erabiliz egiten da. Bezero batek ( $C$ ) eskaera mezua bidaltzen dio denbora zerbitzariari ( $S$ )  $t_0$  unean ( $C$ -ren erlojuarekin neurtua) eta zerbitzariaren erantzuna  $t_1$  unean jasotzen du (baita  $C$ -ren erlojuarekin neurtua). Sarearen ezaugarriak direla eta,  $t_1 - t_0$  gutxienez 1 ms balio duela ezagutzen da. Sinkronizazio saiakera batean,  $C$ -k eskaera sekuentzia bat egiten dio  $S$ -ri, hurrengo taulan agertzen diren balioak lortuz. Erlojuen bereizmena milisegundo batekoa da.

$t_0$	$t_1$	Erantzun mezuan agertzen den $S$ -ren denbora
10:00:03,100	10:00:03,103	10:00:04,341
10:00:03,200	10:00:03,203	10:00:04,442
10:00:03,300	10:00:03,304	10:00:04,541
10:00:03,400	10:00:03,410	10:00:04,645

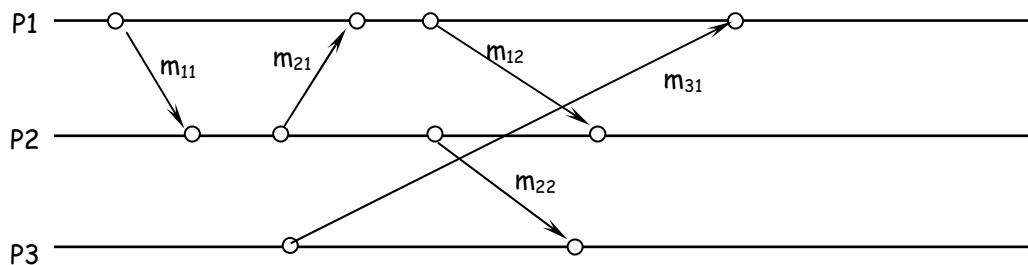
- (a) Emaitzen arabera, kalkula ezazu zein den lor daitekeen prezisiorik hoberena  $C$ -ren erlojuaren sinkronizatzerakoan.
- (b) Kalkula ezazu  $C$ -ren desbiderapena  $S$ -rekiko.
- (c) Azaldu ezazu Cristianen sinkronizazio algoritmoak duen arazoa eskuragarritasuna eta hutsegite tolerantziaren ikuspuntutik. Zentzu honetan, ikusten al diozu abantailaren bat Berkeleyko sinkronizazio algoritmoari? Arrazoitu zure erantzuna.

## 2. ariketa [2,5 puntu]

Hurrengo kronograman sistema banatu bateko hiru prozesuen exekuzioa agertzen da. Prozesuek mezuen bidalketa eta jasoketa gertaerak sortzen dituzte, *bidali(mezua)* eta *jaso(mezua)* hain zuzen.

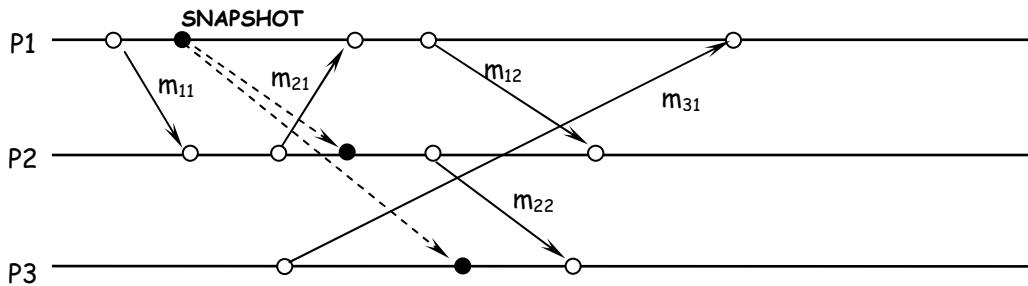


- Kronograman bertan, adierazi ezazu gertaera bakoitzari dagokion denbora logikoa, Lamporten algoritmoaren arabera (hasieran erloju guztiek 0 balio dute):
- Eman kausalki erlazionaturik dauden bi gertaeren adibidea. Baita konkurrenteak diren bi gertaerena ere.
- Adierazi ezazu oraingoan gertaera bakoitzari dagokion denborazko bektorea.



- Marratzu aurreko kronograman sendotasunik ez duen ebaketa bat. Azaldu ebaketa sendotasunik ez duela denborazko bektoreetan oinarrituz.

- (e) Suposatu ezazu P1 prozesuak Chandy-Lamporten algoritmoari hasiera ematen diola, sistemaren egoera global sendo bat kalkulatzeko. Horretarako, SNAPSHOT mezuak bidaltzen ditu (gezi ez jarraiak). (1) Marraztu kronograman Chandy-Lamporten algoritmoaren exekuzioaren jarraipen posible bat, (2) marraztu lortutako egoera globalari dagokion ebaketa, eta (3) adierazi zein izango den lortutako egoera globala, prozesu bakoitzaren eta sei kanalen egoera emanez.



- (f) Posible al da Chandy-Lamporten algoritmoaren exekuzioaren amaieran (kronogramako mezuak eta SNAPSHOT mezuak kontuan hartuta soilik)  $c_{12}$  kanalaren egoera EZ HUTSIK izatea? Zergatik?
- (g) Eta posible al da Chandy-Lamporten algoritmoaren exekuzioaren amaieran  $c_{23}$  kanalaren egoera EZ HUTSIK izatea? Zergatik?

### 3. ariketa [1 puntu]

Bi prozesuko talde bat emanik, P1 eta P2, hurrengo kasu bakoitzeko marraztu ezazu kronograma bat eskatutako hedapen semantikaren adibide bat erakusten duena:

(a) Ez fidagarria

(b) Fidagarria baina ez ordenatua

(c) Fidagarria eta FIFO ordenatua, baina ez kausalki ordenatua

(d) Fidagarria eta guztiz ordenatua, baina ez FIFO ordenatua

(e) Fidagarria eta guztiz ordenatua, baina ez kausalki ordenatua

## 4. ariketa [1 puntu]

Talde-komunikaziorako middleware batek honako operazioak eskaintzen ditu taldearen partaidetzaren kudeaketarako:

*join*( $P_i$ ,  $G$ ).  $P_i$  prozesuaren gehikuntza  $G$  taldean.

*leave*( $P_i$ ,  $G$ ).  $P_i$  prozesuaren kanporatzea  $G$  taldetik.

*install*( $v_{k+1}$ ,  $G$ ).  $G$  taldearen bista berri baten instalazioa.

Middlewareak honako propietateak bermatzen ditu:

- **Bukaera.** Prozesu baten hutsegiteak edota *join()* edo *leave()* operazioek lehenago edo beranduago bista berri baten instalaziora eramaten dute.
- **Akordioa.** Bista bakoitzean, hutsik egiten ez duten prozesu guztiek mezu multzo bera entregatzen dute.
- **Balioetasuna.** Bedi  $P_i$  hutsik egiten ez duen prozesu bat,  $m$  mezua entregatzen duena  $v_k$  bistan. Bista horretako prozesuren bat  $P_j \in v_k$  ez badu  $m$  mezua entregatzen  $v_k$  bistan, orduan  $P_i$  prozesuak instalatuko duen hurrengo bistan  $v_{k+1}$ ,  $P_j$  prozesua ez da azalduko.

Kontsidera ezazu une batean  $G$  taldea hiru prozesuz osaturik dagoela,  $P_1$ ,  $P_2$  y  $P_3$ , eta aurreko propietateen arabera mezuak hedatzen dituztela taldeari:

1. Marraztu kronograma bat  $P_2$  prozesuaren hutsegitea erakusten duena, mezuren bat entregatu gabe. Adierazi ezazu zein operazio exekutatu diren eta noiz instalatuko den bista berria.
2. Hedatu aurreko kronograma,  $P_4$  prozesu berri baten gehikuntza erakutsiz. Adierazi ezazu zein operazio exekutatu diren eta noiz instalatuko den bista berria.

## 5. ariketa [1 puntu]

Aldagai konpartituen atzipena eskaintzen duen zerbitzu batek honako operazioak eskaintzen ditu:

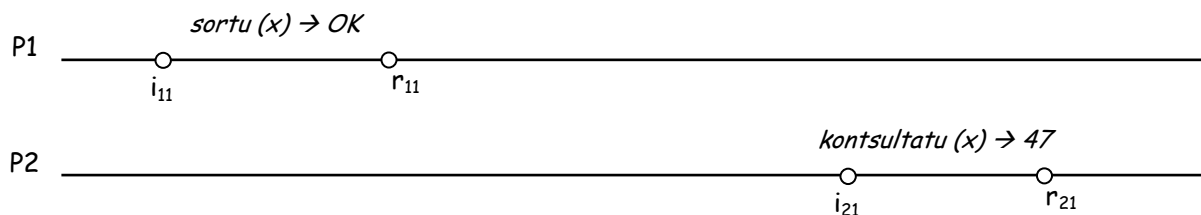
*sortu* ( $x$ ): if (ez da existitzen( $x$ )) {*sortu*  $x$ ;  $x=0$ ; return *OK*} else return *ERROR*;

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

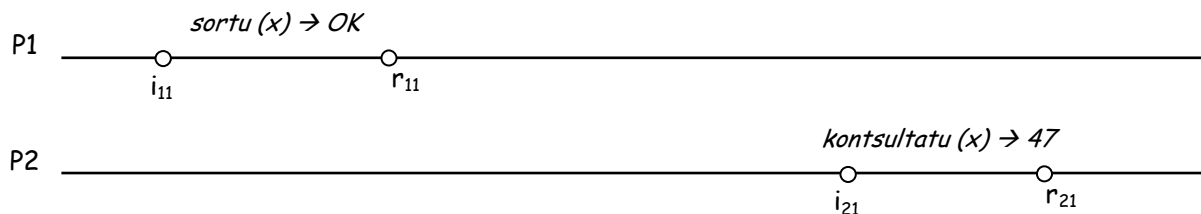
*kontsultatu* ( $x$ ): if (existitzen da( $x$ )) return  $x$ ; else return *ERROR*;

*ezabatu* ( $x$ ): if (existitzen da( $x$ )) {*ezabatu*  $x$ ; return *OK*} else return *ERROR*;

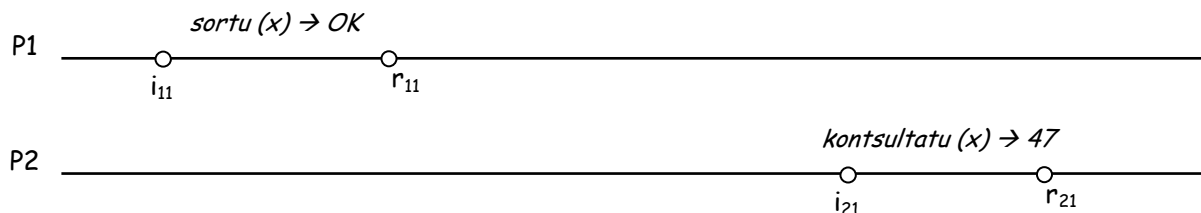
Bedi honako kronograma (osorik ez dagoena), zerbitzu honen bi bezeroen exekuzioa erakusten duena:



- (a) Osatu kronograma bi operazio gehituz, bat *aldatu*() eta bestea *ezabatu*(), azken exekuzioa lerrokagarria izan dadin.
- (b) Egizu orain aurreko gauza bera (*aldatu*() eta *ezabatu*() operazioak gehitu), baina oraingoan azken exekuzioak sendotasun sekuentziala izan dezan baina aldi berean ez dadin lerrokagarria izan.



- (c) Egizu aurreko gauza bera (*aldatu*() eta *ezabatu*() operazioak gehitu), oraingoan azken exekuzioa legala izan dadin, baina ez dezan sendotasun sekuentzialik izan.



- (d) Azaldu bi zerbitzariz osatutako sistema errepikatu baten ezaugarriak eta funtzionamendua, beti lerrokagarriak diren exekuzioak bermatuko dituen. Marraztu sistemaren funtzionamenduaren eskema bat. Adierazi zein behar dituen komunikazioari dagokionez, hau da, ea talde-komunikazio beharrik duen eta zein motakoa.

## 6. ariketa [1,5 puntu]

Bedi egoera gabeko fitxategi zerbitzari baten honako RPCak:

```
status= read_file (ufid, posizioa, buffer, luzera)  
status= write_file (ufid, posizioa, buffer, luzera)
```

*Irakurri/idatzi luzera byte posizioa bytetik hasita, ufid adierazitako fitxategitik. Irakurritako/idatzitako byte kopurua bueltatzen du, edo ERROR.*

- (a) Aldatu aurreko operazioak, egoeradun zerbitzari baterako egokituz. Gehitu itzazu beharrezkoak iruditzen zaizkizun operazioak.

(b) Hasieran emandako operazioak, idempotentek al dira? Eta zuk diseinatutakoak (a) atalean? Arrazoiu zure erantzunak.

(c) Azaldu itzazu egoeradun zerbitzari batek dituen abantailak (1) UNIXeko *unlink* sistema-deiaren semantika eskaintzeko, eta (2) fitxategiei atzipen eskusiboa eskaintzeko (*open* sistema-deian *O\_EXCL* adierazlea erabiliz).

(d) Azaldu zergatik den beharrezkoa fitxategi sistema banatuetan sendotasuna kudeatzeko baliotasun politika bat erabiltzea. Arrazoiu ea egoeradun zerbitzari batek abantailaren bat duen egoera gabekoekin konparatuz (NFS konkretuki) baliotasuna kudeatzeko.