



## Sistema Eragileak II

INFORMATIKA FAKULTATEA

 Konputagailuen Arkitektura eta Teknologia Saila  
 Dpto. de Arquitectura y Tecnología de Computadores

2008ko otsailak 1

<b>Izen-deiturak:</b>	<b>Taldea:</b> Euskara Taldea
-----------------------	----------------------------------

*Estimatutako azterketaren iraupena: 2,5 ordu*

1 [2,5]	2 [2,5]	3 [1]	4 [2,5]	5 [1,5]		
------------	------------	----------	------------	------------	--	--

### 1 Ariketa [2,5 puntu]

Erantzun itzazu ondoko galderak gehienez 3 lerro erabiliz (eta noski letra txikiegia erabili gabe). Lerro gehiagoren erabilerak puntuen galtzea suposatuko du.

**1.a. [0,5 puntu]** FCFS prozesuen planifikazio-politikari dagokionez, jar ezazu adibide bat lau prozesu erabiliz *konboi ondorioa edo efektua*<sup>1</sup> delako arazoa azaltzeko, arazo honek erantzun denboran sortaraz dezakeen kaltea ikus dadin. Adibide berdin horretarako, proposa ezazu ere lau prozesuen exekuzio-ordena egokiena batezbesteko erantzun-denborari dagokionez.

**1.b. [0,25 puntu]** NRU ordezkapen-algoritmoak nahiago izaten ditu orri biktima bezala memoria aldatu gabeko orriak aldatutakoak baino. Zergatik?

---

<sup>1</sup> Ingelesez: *Convoy effect*

**1.c. [0,25 puntu]** Memoriaren kudeaketari dagokionez, zer da *Belady-ren anomalia* izenaz ezagutzen den arazoa, zeinak ordezkapen-politikak desberdintzen dituen arazo hau sortarazi dezaketenen edo ez dezaketenen artean? <sup>2</sup>

**1.d. [0,5 puntu]** Prozesuen sinkronizazio mekanismo guztiak bi taldetan sailkatzen dira: *itxarote aktiboa* eta *itxarotea blokeatuz* bideratzen dutenen artean. Zertan oinarritzen da talde bakoitza? Izenda ezazu talde bakoitzeko mekanismo baten adibidea.

**1.e. [0,5 puntu]** Zein parametro hatu behar dira kontuan interleaving faktore (F) optimoa kalkulatu behar denean? Eta zein da elkarren artean bete behar den erlazioa?

**1.f. [0,5 puntu]** SE berri baterako UNIXen antzeko fitxategi sistema bat definitu nahi da, baina kasu honetan inodoan 10 erakusle zuzen, eta 1. eta 2. mailako zeharkako erakusleak egongo dira soilik, eta ez 3. zeharkako erakuslerik. Suposatuz gure blokeen tamaina 2KBytekoa dela eta bloke erakusleak 4 bytekoak; kalkula ezazu zein izango den fitxategi baten tamaina maximoa.

---

<sup>2</sup> Laguntza: arazo hau sortarazten ez duten ordezkapen politikak *pila-algoritmo* izenaz ezagutu ohi dira.

## 2 Ariketa [2,5 puntu]

Konkurrentzian exekutatzen diren ondoko prozesuen kodea eman digute:

```
A programa           B programa           C programa
{
  (2 CPU tick)
  jaitsi (sem_b1)
  (1 CPU tick)
  (1 tick blokeatua)
  (1 CPU tick)
  jaitsi (sem_b3)
  (1 CPU tick)
  (7 tick blokeatua)
  (1 CPU tick)
  igo (sem_b1)
  (5 tick blokeatua)
  igo (sem_b3)
  (1 CPU tick)
}

{
  (2 CPU tick)
  jaitsi (sem_b2)
  (1 CPU tick)
  (5 tick blokeatua)
  (2 CPU tick)
  jaitsi (sem_b1)
  (1 CPU tick)
  igo (sem_b2)
  (2 CPU tick)
  igo (sem_b1)
}

{
  (2 CPU tick)
  jaitsi (sem_b3)
  (5 CPU tick)
  jaitsi (sem_b2)
  (5 CPU tick)
  igo (sem_b2)
  (2 CPU tick)
  igo (sem_b3)
  (1 CPU tick)
}
```

Goiko diagraman kode atal bakoitzeko adierazten da exekutatze behar den denbora edo blokeatuta egongo den denbora. Demagun ondoko prozesuak dauzkagula:

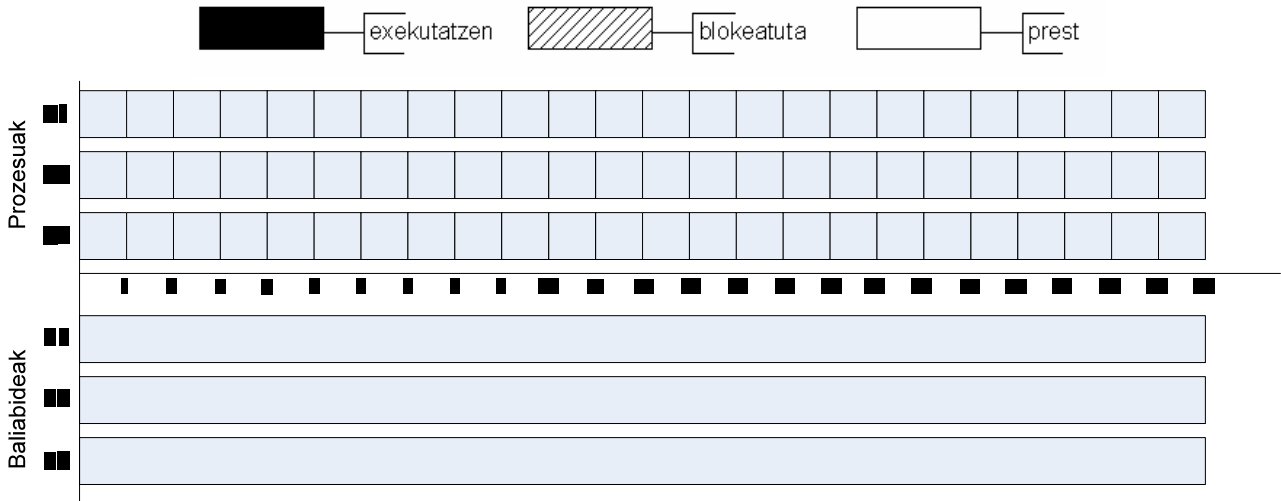
<u>Prozesua</u>	<u>Programa</u>	<u>Lehentasuna</u>
<b>P1</b>	<b>A_programa</b>	<b>3</b>
<b>P2</b>	<b>B_programa</b>	<b>2</b>
<b>P3</b>	<b>C_programa</b>	<b>1</b>

### Oharrak:

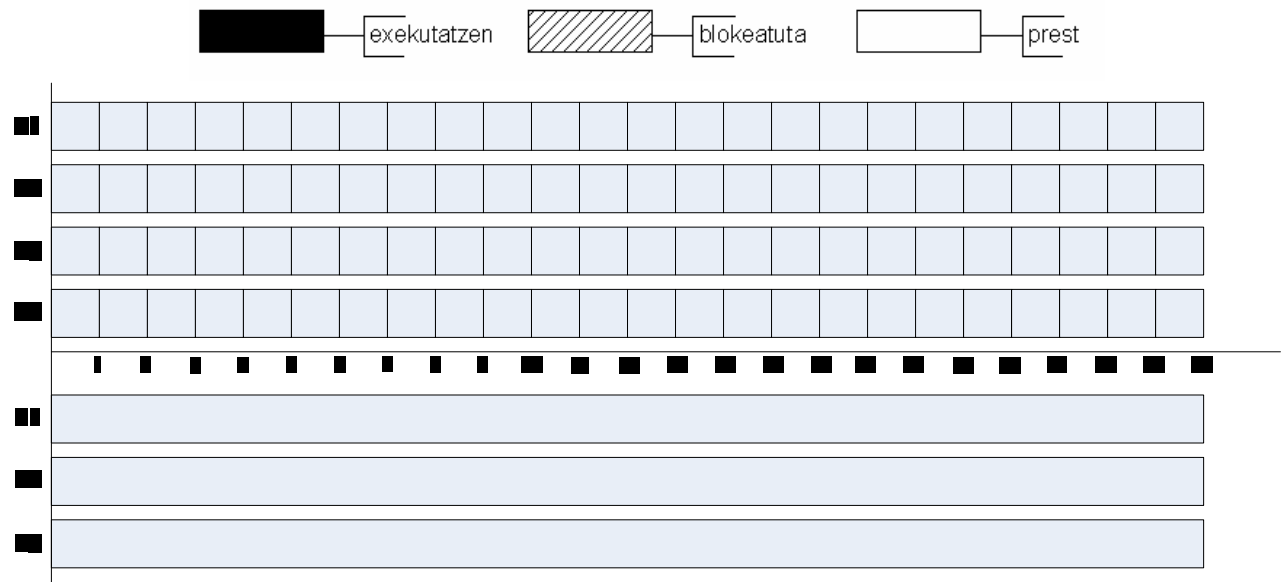
- kontsideratu lehentasun-maila handiagoak lehentasun handiagoa adierazten duela
- Suposatu semaforoaren **igo** eta **jaitsi** eragiketen exekuzio-denborak oso txikiak direla. Beraz, denbora horiek ez dira kontutan hartuko, eta aurreko CPU denboraren barruan sartzen dela suposatuko dugu. Adibidez, **B\_programa**-k hasieran exekutatzen dituen 2 CPU tick-etan barneratuko da semaforoaren `jaitsi(sem_b2)` den hurrengo agindua.
- Round Robin erabiltzen den kasurako, desblokeaketa bat eta quantum agorpen bat gertatzen badira une berdinean, desblokeaketa lehenago tratatuko dugu.
- `igo(sem_bx)` agindua exekutatzen ari dela, prozesu batek quantum-a agortzen badu, suposatuko dugu agindua quantum-a agortu baino lehen osorik exekutatu duela.

Baldintza hauek kontsideratuz ondokoa eskatzen da:

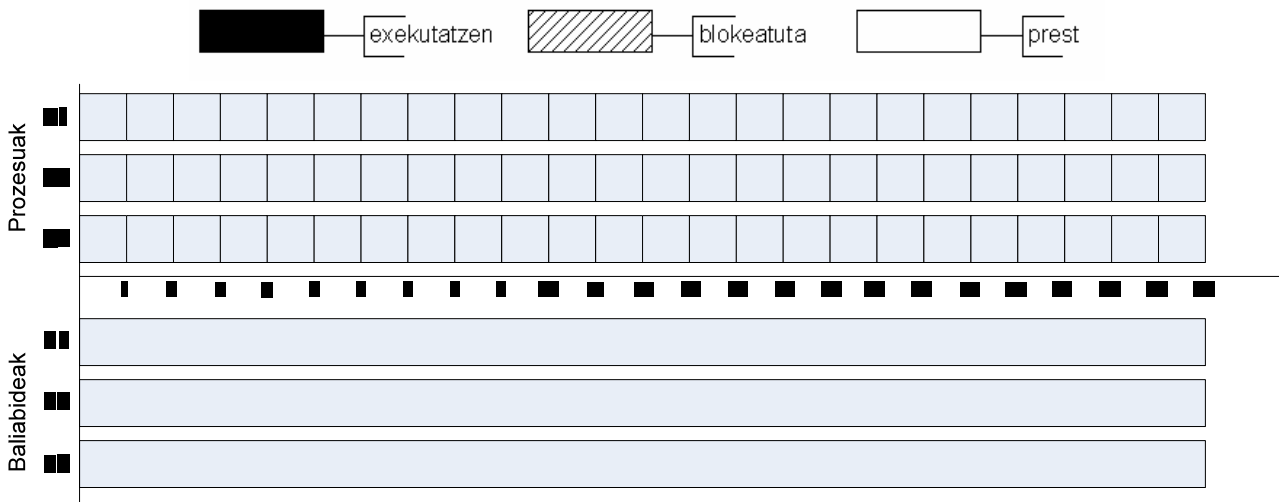
2.a. Marraz itzazu ondoko exekuzio-kronogramaren lehendabiziko 24 tick-ak ondoko planifikazio politikarako: **FCFS, Round Robin (q=3), eta gertaera bidezko kanporaketa.** Suposatu hiru prozesuak 0.tick-ean hasi direla denak elkarrekin eta P1, P2 eta P3 ordenan iritsi direla.



2.b. Marraz itzazu ondoko exekuzio-kronogramaren lehendabiziko 24 tick-ak ondoko planifikazio politikarako: **Lehentasun estatikoak eta gertaera bidezko kanporaketa.** Suposatu hiru prozesuak 0.tick-ean hasi direla denak elkarrekin eta P1, P2, P3 ordenan iritsi direla.



2.c. Marraz itzazu ondoko exekuzio-kronogramaren lehendabiziko 24 tick-ak ondoko planifikazio politikarako: **Lehentasun estatikoak, Round Robin (q=3), eta gertaera bidezko kanporaketa**. Suposatu hiru prozesuak 0.tick-ean hasi direla denak elkarrekin eta P1, P2, P3 ordenan iritsi direla.



2.d. Kalkulatutako hiru kronogramen arabera, aurreko planifikazio-politiken artetik zein da egokiena sistemaren CPU-tasari dagokionez? Arrazoitu zure erantzuna.

2.e. Zein izango da planifikazio-politika guztietatik egokiena batezbesteko erantzun-denborari ( $t_r$ ) dagokionez? Arrazoitu zure erantzuna.

2.f. Aurreko (2.a.) ataleko kronogramari dagokionez, kalkula ezazu eraginkortasunaren galera ( $1 - E_f$ ) testuinguru aldagetak direla eta, kontuan izanik 1.400 MHz-etako prozesadore bat dugula, erlojuaren maiztasuna 100 Hz dela, eta testuinguru-aldaketa bakoitza burutzeko denbora ( $t_{cs}$ ) 1 ms dela.

### 3 Ariketa [puntu 1]

Aurreko ariketan emandako programa hauek kontuan izanik:

```
A programa
{
  (2 CPU tick)
  jaitsi (sem_b1)
  (1 CPU tick)
  (1 tick blokeatua)
  (1 CPU tick)
  jaitsi (sem_b3)
  (1 CPU tick)
  (7 tick blokeatua)
  (1 CPU tick)
  igo (sem_b1)
  (5 tick blokeatua)
  igo (sem_b3)
  (1 CPU tick)
}

B programa
{
  (2 CPU tick)
  jaitsi (sem_2)
  (1 CPU tick)
  (5 tick blokeatua)
  (2 CPU tick)
  jaitsi (sem_1)
  (1 CPU tick)
  igo (sem_2)
  (2 CPU tick)
  igo (sem_b1)
}

C programa
{
  (2 CPU tick)
  jaitsi (sem_b3)
  (5 CPU tick)
  jaitsi (sem_2)
  (5 CPU tick)
  igo (sem_b2)
  (2 CPU tick)
  igo (sem_b3)
  (1 CPU tick)
}
```

**3.a.** Aipatu elkarblokeaketa egoera gertatuko litzatekeen kasuren bat, kodean argi azalduz zein aginduan geratuko litzateke prozesu bakoitza. Halaber, marraztu baliabideen asignazio-grafo baten bidez elkarblokeaketa egoera hau nola detektatuko genukeen.

**3.b.** Kode honetan elkarblokeaketak gertatzeko arriskua dagoenez, eta suposatuz sistema eragileak *ostrukaren algoritmoa* erabiltzen duela arazo honi aurre egiteko, zein soluzio proposa dezakezu programa hauen kodean arazo hau ekidin ahal izateko?

#### 4 Ariketa [2,5 puntu]

Memoria orrikatua duen memoria-sistema bat dugu, ondoko ezaugarriak dituena: 32 biteko helbideak dituen alegiazko memoria, eta 4 Kbyteko orriak. Sistema eragile hau diseinatu da gehienez 32 Gbyte RAM helbideratu ahal izateko gehienez, baina kasu honetan gure ordenadoreak 512 Mbyteko memoria fisikoa du.

Guzti hau kontuan izanik, erantzun itzazu ondorengo galderak, erantzunak arrazoituz:

**4.a.** Marraz itzazu helbide logiko eta fisikoaren eskemak.

**4.b.** Gure ordenadore honetan bit-mapa erabiltzen bada tarte librearen kudeaketarako, zein izango da bere tamaina bytetan?

**4.c.** Zein da memoria kudeaketa-sistema honen batezbesteko kanpo-fragmentazioa prozesu bakoitzeko? Arrazoitu zure erantzuna.

**4.d.** Eskema honetan exekutatu diren prozesuak tamaina berdinekoak direla jakinik, non prozesu bakoitzaren working-setaren tamaina 256 orrikoa izango den, zenbateko izango da multiprogramazio-maila maximoa gure ordenadore honetan?

**4.e.** Gure hardwarea hobetzen badugu RAM memoria handituz 1Gbyte RAM izateko, aurreko ataletako zein parametro handitu edo hobetuko lirateke? Listatu besterik ez dituzu egin behar adieraziz handitu ala txikitu egiten diren, ez dituzu berriro kalkulatu beharrik.



**4.f.** Sistema orrikatu honen ondoko orri-taulen definizioa emanik:

```
struct sarr_ot {                                /* OT baten sarrera bakoitza */
    BIT V;                                      /* baliagarritasun bita */
    BIT R;                                      /* erreferentzia bita */
    int KargaTick;                             /* orria zein tick-ean kargatu den
                                              memorian */
    int markoa;                                 /* marko zenbakia */
} OT [PROZ_KOP][OT_LUZERA]; /* prozesu guztien orri-taulak */
```

eta parametro bakar bezala prozesu baten identifikadorea duen ondoko memoriaren kudeaketarako errutina izanik

```
int OrdezkapenPolitika (int pid)
{
    int i, biktima=-1, markoa=-1, txikiena=MAXINT;
    for (i=0; i<OT_LUZERA; i++) {
        if ((OT[pid][i].V == 1)&&(OT[pid][i].KargaTick<txikiena)) {
            biktima = i;
            markoa = OT[pid][i].markoa;
            txikiena = OT[pid][i].KargaTick;
        }
    }
    return markoa;
}
```

Ondoko galderak erantzun itzazu era laburrean:

1. Klasean ikusitako zein ordezkapen algoritmoa da bertan implementatzen dena?  
Arrazoitu zure erantzuna.
  
2. Zein da V bitaren esanahia? Zertarako erabiltzen da V bita ordezkapen algoritmo honetan?

3. Zein da ordezkapen algoritmo honen asignazio eremua, lokala ala globala?  
Arrazoitu zure erantzuna.

**4.g.** Aldatu ezazu aurreko azpiataleko kodea, orri-biktima bilatzeko ordezkapen-algoritmoa NRU (Not Recently Used) izan dadin dagoenaren orde. Horretarako, erantsi itzazu nahiz kodean edo orri taularen definizioan beharrezkoak deritzozun definizio berriak.

## 5 Ariketa [1,5 puntu]

Hurrengo diagraman FAT16 fitxategi-sistema erabiltzen duen gure disko unitate baten katalogo eta FAT taularen sarrera batzuk erakusten dira. Fitxategi-sistema hau ondoko ezaugarriak dituen disko gogorrean ezarri da: 256 zilindro, 8 pista/zilindro, eta 1 KBytetako 128 sektore. Diskoaren bira abiadura 7200 bira/min.-koa da, eta diskoaren irakurketa/idazketa buruak batez beste 30 mseg behar ditu zilindro batean kokatzeko. DMAk 190 µseg behar ditu sektore bat memoriara transferitzeko, eta disko kontroladorearen datu-bufferraren tamaina 1 Kbytekoa da. Bloke logikoak 4KBytekoak dira.

**KATALOGOA**

Izena	Atributuak	1. blokea	Luzera
Bezeroak	...	10	7
Produktuak	...	9	5
	...		
	...		

**FAT**

0	
1	
2	
3	5
4	11
5	7
6	3
7	18
8	49
9	12
10	6
11	EOF
12	16
13	19
14	EOF
15	13
16	4
17	42
18	14
19	21
20	

Ondokoa eskatzen da:

**5.a.** Kalkula ezazu disko unitatearen edukiera maximoa

**5.b.** Beharrezkoa al da diskoa partiziotan banatzea sistema eragileak bere edukiera osoa helbideratu ahal izan dezan? Baiezko kasuan, nola zatituko zenuke?

**5.c.** Kontsideratuz disko honen kontroladoreak 3 faktoreko interleaving-arekin formateatu dela, kalkula ezazu bloke bat atzitzeko behar den denbora.

**5.d.** Ezaguna da gure "Bezeroak" fitxategia 256 bytetako erregistro logikoetan antolatuta dagoela. 22. erregistro logikoa atzitu nahiko bagenu, zein da jarraituko litzatekeen helbideen itzulpen prozesua S/I errutinatik abiatuta disko-gestoreraino? Adieraz itzazu zeintzuk izango diren bloke erlatibo, bloke absolutu, bloke fisiko eta zilindro-pista-sektore balioak<sup>3</sup>.

---

<sup>3</sup> Azken kalkuletarako ondoko eragiketak erabil ditzakezu behar izanez gero: **sektorea**= $b\%S$ ; **zilindroa**= $b \operatorname{div} (S*P)$ ; **pista** =  $(b\%(S*P)) \operatorname{div} S$ ; non S= sektore kopurua, P=pista kopurua, eta b=bloke zenbakia