



Izen-deiturak:	Taldea: Euskara Taldea
-----------------------	----------------------------------

Arren, azterketa amaitzen duzunean, dagokizun amaiera orduaren gelaxka marka ezazu mesedez.	9:00 9:15	9:15 9:30	9:30 9:45	9:45 10:00	10:00 10:15	10:15 10:30	10:30 10:45	10:45 11:00	11:00 11:15	11:15 11:30	11:30 11:45	11:45 12:00
---	--------------	--------------	--------------	---------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

Estimatutako azterketaren iraupena: 3 ordu

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

1 Ariketa [3 puntu]

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

1. Deskriba eta bereiz itzazu esaldi bana erabiliz *programa*, *prozesua* eta *thread* kontzeptuak.

programa:

prozesua:

thread:

2. Elkarblokeaketaren arazoari aurre egiteko ikasgaiari ikusitako mekanismo posibleen artean ondoko biak ditugu: *predikzioa* eta *prebentzioa*. Zein da bien arteko desberdintasuna?

3. Diskoaren kudeaketari dagokionez, *interleaving* izeneko teknika proposatu da; Zertan datza teknika hau?

4. Disko bloke logiko baten atzipena egiteko behar den denbora kalkulatu nahi dugunean, zeintzuk dira denbora honetan kontuan hartu behar diren dispositiboaren eragiketa guztiak?

5. Zer da IORB bat? Zertarako erabili ohi da? Nola erabili ala aplikatzen dira hauek dispositibo baten driverrarean?

6. Sistema bateko bloke logikoaren tamaina 4 Kbytekoa izanik, eta 15Kbytetako F1 fitxategi bat emanik, adieraz ezazu grafikoki nola antolatuko liratekeen fitxategi honi dagozkion datu-blokeak ondoko fitxategi sistemak emanik:
 - a. Unix
 - b. FAT32

Oharra: suposa ezazu disko horretan K1 katalogo bakar bat dagoela, zeinetan fitxategi hau dagoen

2 Ariketa [1 puntu]

Gure aplikazio batean ekoizle-kontsumitzaile eskema inplementatu nahi dugu, komunikazioa bideratuz buffer konpartitu bat erabiliz. Horretarako, bi prozesu mota dauzkagu; ekoizleak eta kontsumitzaileak, ondoko bi programak idatzi ditugularik:

<pre>/* ekoizlea */ info_mota elementua; { while (1) { ekoiztu(&elementua); idatzi_buffer(elementua); } }</pre>	<pre>/* kontsumitzailea */ info_mota elementua; { while (1) { irakurri_buffer(&elementua); kontsumitu(elementua) } }</pre>
---	--

idatzi_buffer() eta irakurri_buffer() eragiketak aurreprogramatuta dauzkagu eta ez dute atzipen eksklusiboa bideratzen komunikazio-buffer baten gainean. Komunikazio-buffer honek gehienez info_mota motako N elementu gorde ditzake. Aurreko kodean argi ikus daitekeenez, ez da inolako sinkronizazio mekanismorik erabiltzen buffer hutsa eta betearen baldintzak kudeatzeko. Ondokoa eskatzen da: aurreko bi programen kodea alda ezazu semaforoak erabiliz sinkronizazio behar guztiak bete daitezen.

<pre>/* Hasieraketak */</pre>	
<pre>/* ekoizlea */ info_mota elementua; { while (1) {</pre>	<pre>/* kontsumitzailea */ info_mota elementua; { while (1) {</pre>
<pre> } }</pre>	<pre> } }</pre>

3 Ariketa [2,5 puntu]

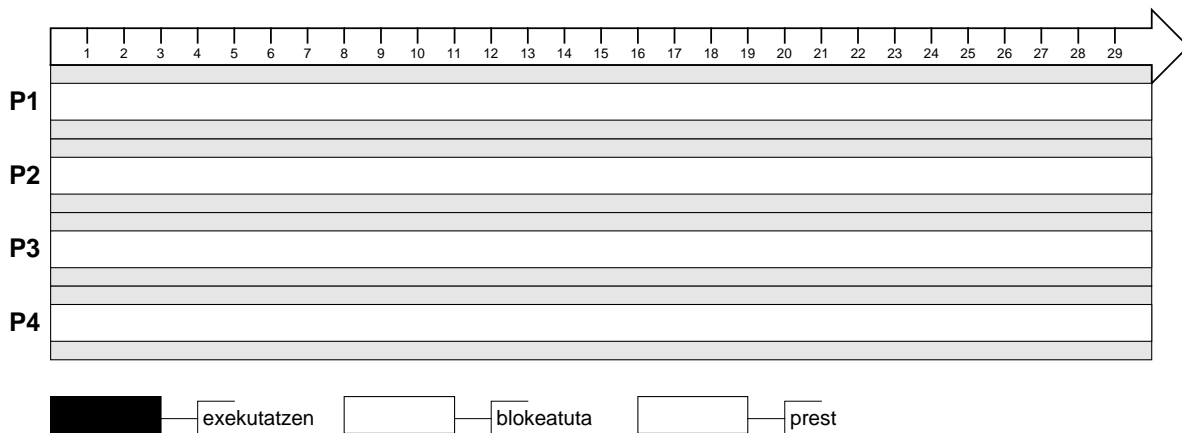
Sistema batean exekutatu diren lau prozesu desberdinen informazioa daukazu ondoko taulan. Bertan dagoen informazioa ondokoa da: prozesuaren identifikadorearekin batera sorrera tick-a, lehen eta bigarren CPU tarteen denbora tick-etan, aldiro blokeatuta egongo diren denbora, eta beren lehentasuna (balio handiago batek lehentasun maila handiagoa adierazten du).

Prozesu guztiak antzeko jokaera erakusten dute: lehenengo CPU tartea igaro ondoren blokeatu egiten dira, ondoren bigarren CPU tartea exekutatu eta azken hau burutzerakoan amaitu egiten dira (adib. P1-k 6 CPU tick exekutatu ondoren, blokeatu egingo da 2 tick-etan zehar, eta beste 6 tick exekutatu ondoren amaitu egiten da). Baldintza berdinen aurrean bi prozesu edo gehiago aukera badaitezke, *schedule* identifikadore (*pid*) txikiena duena aukeratuko du.

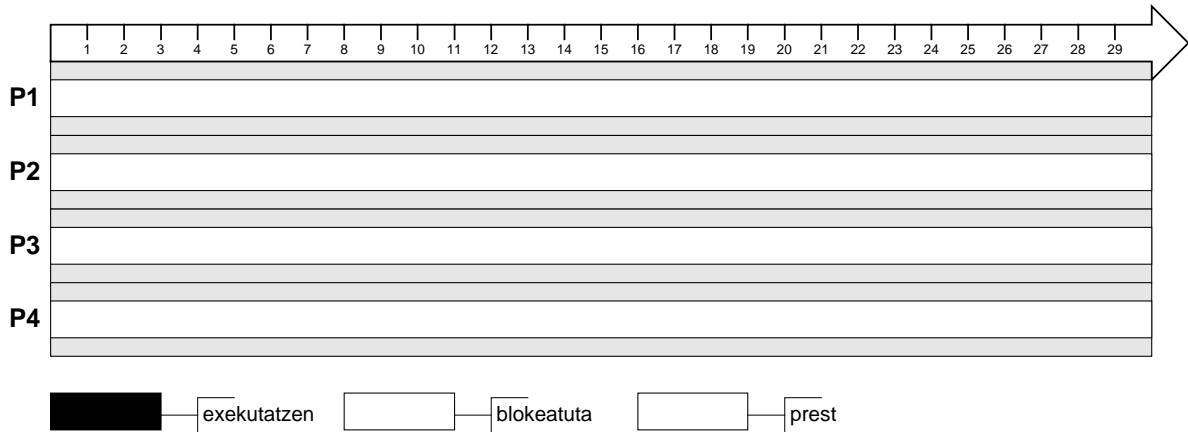
Prozesua	Sorrera tick	Lehenengo CPU tartea	Denbora blokeatuta	Bigarren CPU tartea	Lehentasuna
P1	0	6	2	6	1
P2	2	2	6	2	3
P3	0	4	3	3	2
P4	4	1	5	2	3

Ondokoa eskatzen da:

- a) Marraz ezazu exekuzio diagrama prozesuen planifikazio politika lehentasun estatiko bidezkoa denean, gertaera bidezko kanporaketa burutzen delarik prozesu bat *prest* egoeraren ilarara iristen den bakoitzean. Lehentasun berdineko prozesuen artean FCFS politika jarraituko da.



- b) Marraz ezazu ondoko beste diagraman zein exekuzio diagrama izango genukeen *Round Robin* konbinatzen dugunean aurreko lehentasun kanporatzailearen politikarekin. Horretarako kontsideratu behar duzu sistema honetan prozesu guztientzako *quantum* balioa berdina dela, $q=1$ izanik.



- c) Kalkula ezazu bi kasuetarako, eta prozesu bakoitzeko, batezbesteko erantzun denbora (latentzia, t_r), batezbesteko amaiera-denbora (t_f), eta prozesu bakoitzaren CPU-tasa (t_{CPU}). Horretarako ondoko taula bete behar duzu.

	a)			b)		
	t_r	t_f	CPU tasa	t_r	t_f	CPU tasa
P1						
P2						
P3						
P4						
batez- bestekoa						
Sistema-ren CPU tasa						

4 Ariketa [2 puntu]

Memoria segmentatu-orrikatua duen sistema batek 4 Gbyte helbideratu ditzakeen helbide logikoak dauzka, eta memoria erreal 512 Mbytekoa da. Kontuan izanik 16 Kbyteko orriak erabiltzen direla, eta prozesu bakoitzak gehienez 16 segmentu izan ditzakeela, ondoko galderak erantzun itzazu:

a. Alegiazko helbide eta helbide fisikoaren egitura

b. Segmentu-taularen tamaina

c. Orri-taularen tamaina

d. Prozesu baten segmentu- eta orri-taulen tamaina, prozesuaren helbidera dezakeen memoria tamaina maximoa lortzeko.

e. Prozesu bakoitzak izan dezaken fragmentazio maximoa.

5 Ariketa [1,5 puntu]

Disko batek 80 zilindro, 2 alde, eta 40 sektore pista bakoitzeko dauzka. Honen gainean blokeak atzitzeko bi politika desberdin artean bat aukeratu nahi da; SCAN eta C-SCAN. Diskoaren irakurri/idazteko buruaren abiadura batez beste 1 mseg zilindro bakoitzekoa da, eta birkalibratzeko eragiketaren denbora 45 mseg. Kontuan izanik une konkretu batean burua 45. pistan dagoela (33.a atzitu izan eta gero),

a) erabaki ezazu zein den bi politika hauetatik hoberena ondoko pistak atzitzeko eska sekuentziari dagokionez: 2,5,23,34,47,55,36,28,12,3

b) SCAN eta C-SCAN politiken artean aukeratzean, bakoitzak eskaerak zerbitzatzeko behar duen denboraz gain, atzitzen den informazioaren kokapenarekin zerikusia daukan beste irizpide bat kontutan izan behar dugu. Zein da azken irizpide hau?

c) SSF (gertuen dagoena) politika ere erabili liteke planifikazio-politika bezala, baina errealitatean ez da aplikatzen. Zein arazoa dela eta ez da aplikatzen errealitatean politika hau?