



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: 2 ordu t'erdi

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

1 Ariketa [2 puntu]

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

1.a. [0.25 puntu] Gauza berdina al dira elkar-blokeaketa (*dead-lock*) eta itxarote mugagabea? Arrazoitu zure erantzuna.

1.b. [0.25 puntu] Gerta al daiteke *bidali* mezu-trukerako primitiboa erabiltzerakoan bidalketa egiten duen prozesua blokeatuta geratzea? Baiezko kasuan, azal ezazu zein egoeran gerta daitekeen halakorik.

1.c. [0.75 puntu] Ondoko taulan fitxategi baten atzipena egiteko jasaten diren helbide itzulpeneko datu-aldaketa guztiak azaltzen dira. Ondokoa eskatzen da: taula bete ezazu argi azalduz pausu bakoitzean zein den itzulpena egiteaz arduratzen dena (S/I sistema deiaren errutina, disko gestorea, erabiltzaile mailako aplikazioa, diskoaren kontroladorea...), itzulpen hori egiterakoan exekuzio modua erabiltzaile mailakoa (E) ala sistema edo kernel modukoa (S) den, eta zein datu-egitura erabiltzen dituen lan hau egin ahal izateko (dispositiboaren deskribatzailea, FAT, kanal-taula, hash taula...).

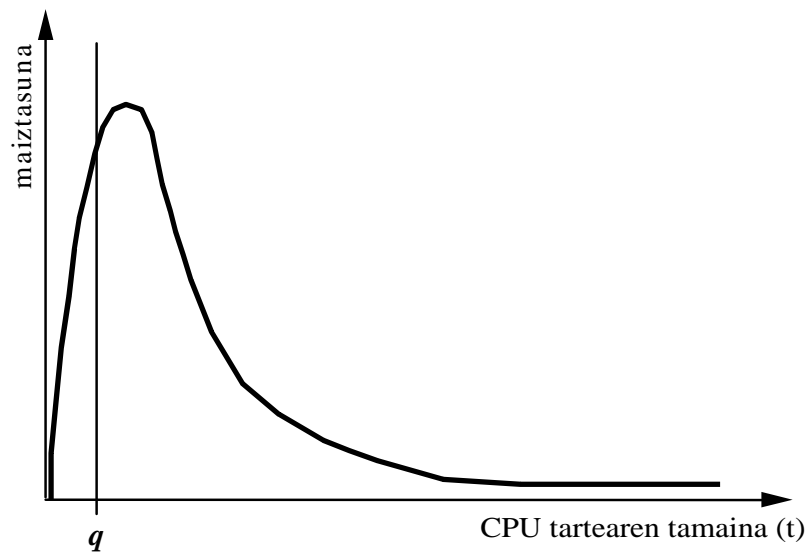
Helbide-itzulpenaren unekak	Nork egiten duen	Exekuzio modua (erabiltzaile/sistema)	Erabilitako datu-egiturak
Antolaketaren kudeaketa (erregistro logiko zenbakiaren kalkulua)			
Erreg. logiko → Sistema-deien parametroak			
Sistema-deien parametroak → Bloke logiko erlatiboa			
Bloke logiko erlatiboa → Bloke logiko absolutua			
Bloke logiko absolutua → Bloke fisiko			
Bloke fisiko → zilindro, pista, sektore balioak			
Software interleaving-aren kudeaketa			
Hardware interleaving-aren kudeaketa			

OHARRA: Taula bete ahal izateko, ondoko zerrendak erabil ditzakezu besteak beste, nahiz zenbait kasutan ezer ez den idatzi behar edo beste aukera desberdinak ere posible diren:

- 2.zutabea: 1-S/Iko sistema deiaren errutina, 2-Disko gestorea, 3-Erabiltzaile mailako aplikazioa, 4-Diskoaren kontroladorea, 5-Swapperra, 6-epe laburreko schedulerra...
- 3. zutabea: E ala S.
- 4. zutabea: a-Dispositiboaren deskribatzailea, b-FAT, c-Kanal-taula, d-Hash taula, e-Inodoen taula, f-inodoa, g-Disko unitatearen superblokea, h-Katalogoa,...

1.d. [0.25 puntu] Aurreko ataleko adibidean, ezer aldatuko al litzateke atzitu nahi den fitxategia CD unitate batean edo disko gogor batean balego? Arrazoitu zure erantzuna. Aldatu beharreko kasuan, aldatu beharreko atalak zeintzuk liratekeen azal ezazu.

1.e. [0.5 puntu] Sistema bateko errendimenduaren hobekuntza lortzeko asmoz, esperimentu bat egin dugu jakiteko zer nolakoak diren bertan exekutatzeko CPU tartea. Esperimentu hau ordenadore bakar batean egin da bertan exekutatzeko aplikaziorik arruntenak aukeratuz. Emaitzak ondoko diagraman azaltzen dira, bertan azaltzen delarik ere zein den gure sistemaren quantum-aren balioa (q).



Emaitza hauek kontuan izanik, erantzun itzazu ondoko galderak:

1.e.1. Quantumaren balioa egokiena dela al deritzozu? Horrela ez bada, zer egingo zenuke hau konpontzeko?

1.e.2. Gure makina honetan exekutatu nahi diren aplikazioak, esperimentua egindako prozesuak baino kalkulura zuzendutakoagoak balira, orain arte izandako quantumaren balioa egokia izango litzatekeela uste al duzu? Arrazoitu zure erantzuna

2 Ariketa [1,5 puntu]

Memoriaren kudeaketarako eredu segmentatu-orrikatua darabilen sistema batean orriak 4 Kbytekoak dira eta segmentu-taula 256 elementuk osatzen dute gehienez. Prozesuen helbideratze-tartea 32 bitekoa da. Memoria sistema hau 128 Mbyteko RAM memoria duen ordenadore batean ezarri da. Zera eskatzen da:

2.a. Helbide logiko eta fisikoen eskemak.

2.b. Orri-taula bakoitzaren osagaien kopurua (zenbat sarrera dituen taulak).

2.c. Segmentu bakoitzeko dagoen orri-taulen kopuru maximoa.

2.d. Segmentu baten tamaina maximoa (bytetan).

2.e. Memoriaren eredia orrikatu sinplea izan balitz (orrikapena maila bakar batean), helbideratze tartea eta orrien tamaina aldatu gabe, zein izango litzateke orri-taularen tamaina maximoa?

2.f. Memoria librearen kudeaketa bit-maparen bidez egingo balitz, zein izango litzateke bit-maparen tamaina bytetan ordenadore honetan?

2.g. Zein izango litzateke sistema honen kanpo fragmentazioaren batezbesteko estimazioa sistema segmentatu-orrikatuan? Eta sistema orrikatu hutsean?

2.h. Sistema honetan zein izango da multiprogramazio-maila maximoa prozesu bakoitzeko working-set tamaina minimoa 4 orrikoa bada.

3 Ariketa [2,5 puntu]

Linuxen Java programazio-lengoaia erabiltzen dugu, baina Windows 98tik atzitu nahi ditugu gure datuak. Zoritxarrez, Linuxerako fitxategi sistemako disko batean dauzkagu fitxategi guztiak, eta beraz informazio guztia FAT-ean oinarritzen den fitxategi sistema duen beste disko batera kopiatu beharko dugu. Bi diskoak tamaina berdinekoak dira: 2 Gbytetakoak. FAT16 fitxategi sistema aukeratu dugu Windowserako.

Linuxen kasuan blokearen tamaina 2Kbytekoa da eta blokearen erakusleak 4 bytekoak.

Java lengoaiaren ezaugarri jakina da fitxategi ugari eta tamaina oso txikikoak dituela. Oro har, ondoko fitxategiak aurkitu ditugu gure Linux diskoan:

- A. 2 Kbyte baino txikiagoak diren 2048 fitxategi. Batezbesteko tamaina bezala 1Kbyte kontsideratuko dugu
- B. 1024 fitxategi 2Kbyte baino handiagoak. Batezbesteko tamaina bezala 1Mbyte kontsideratuko dugu

3.a. Kalkula ezazu zein den FAT16 fitxategi sisteman erabili beharreko blokearen tamaina 2Gbyte osoak helbideratu ahal izateko.

3.b. Zein da bi fitxategi sistemetan gorde daitekeen fitxategirik tamaina handiena?
FAT16:

Linux:

3.c. Zein da gure FAT16 diskoan fragmentazioa dela eta galduko den tartea?

3.d. Zein da Linux diskoan izango den batezbesteko fragmentazioa?

3.e. Orain arte kalkulatuakoren arabera, zenbatekoa izango da FAT16 diskoan egongo den tarte librea? Eta Linuxeko fitxategi sistema duenean? (arbuiagarri bezala kontsidera ezazu FAT, katalogoak, inodoak eta indize-blokeetan galdutako disko-tartea).

3.f. Aurreko azpiataletako emaitzen aurrean, zein fitxategi-sistemak erakusten du errendimendu hobea fragmentazioari begira? FAT32 fitxategi sistema erabili izan balitz emaitzak desberdinak izango al lirateke? Zergatik? Arrazoitu zure erantzuna datu zehatzekin.

4 Ariketa [2 puntu]

Prozesuen hiru planifikazio politika konpara ditzazun eskatzen da. Horretarako, sekula blokeatzen ez diren ondoko prozesuen sorrera unea, estimatutako CPU denbora (balioak tick-etan adierazten dira), eta lehenasuna ematen dira (balio handienak lehenasun maila handiena adierazten du).

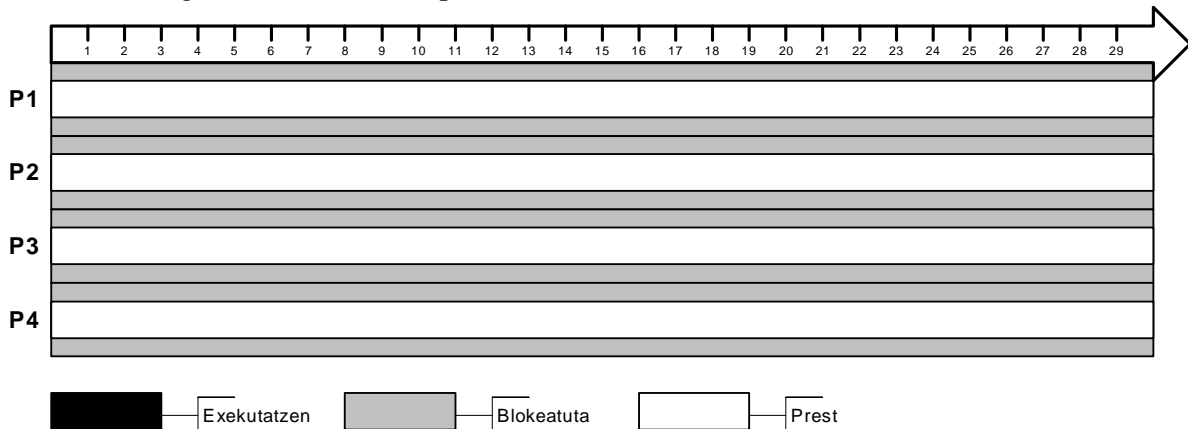
Prozesua	Sorrera-tick	Lehenasuna	Estimatutako CPU denbora
P1	0	2	14
P2	4	3	7
P3	6	4	3
P4	7	4	6

Konparatu behar diren hiru planifikazio politikak ondokoak dira: SJF gertaera bidezko kanporaketarekin, FCFS eta *Round-Robin* (3 tickeko *quantum*-arekin), eta lehenasun kanporatzailea denbora eta gertaera bidez ($q=2$ izanik beste kasu honetan).

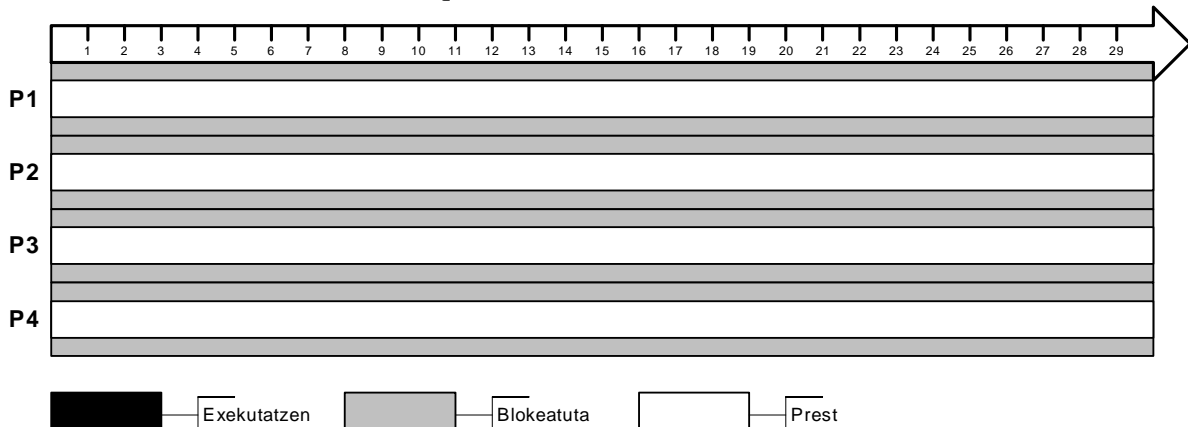
Ondokoa eskatzen da:

4.a. Osa itzazu ondoko kronogramak, eskatutako hiru planifikazio politiketarako.

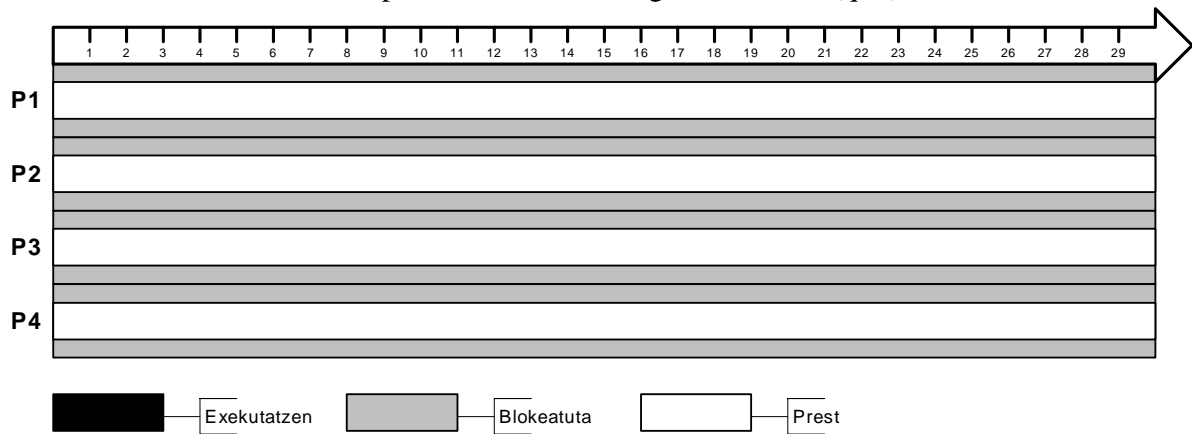
4.a.1. SJF gertaera bidezko kanporaketarekin.



4.a.2. FCFS eta *Round-Robin* ($q=3$).



4.a.3. Lehentasuna eta kanporaketa denbora eta gertaera bidez ($q=2$).



4.b. Zenbat testuinguru aldaketa sortarazi dira hiru planifikazio politika hauekin?

4.c. Zenbatekoa da latentzia hiru politika hauetan?

4.d. Hiru politika hauetatik zein da zure ustez CPU-tasa egokiena lortzen duena? Arrazoitu zure erantzuna datu zehatzak azalduz.

5 Ariketa [2 puntu]

a) Sistema orrikatu batean ordezkapen algoritmo desberdinak aztertu nahi ditugu. Jo dezagun 9 marko dituen kasu konkretu bat dugula, non ondoko ezaugarriak ditugun marko bakoitzean:

Markoa	<u>Karga-tick</u>	<u>V bita</u> <u>(baliagarri-</u> <u>tasun bita)</u>	<u>R bita</u> <u>(erreferentzi</u> <u>bita)</u>	<u>M bita</u> <u>(aldatuaren</u> <u>bita)</u>	<u>Erreferentzien kontagailu,</u> <u>erref. bufferra, ala</u> <u>desplazamendu erregistroa</u>
0	1290	1	0	1	9678
1	1300	1	1	1	9576
2	7489	0	0	0	0
3	1304	1	1	1	9701
4	1298	1	0	0	9685
5	1277	1	1	0	9694
6	1310	1	0	1	9699
7	1299	1	1	1	9697
8	1301	1	1	0	9690

Ordezkapen politika guztiak eremu globalekoak direla kontsideratuz, adieraz ezazu zein izango litzatekeen aukeratutako orri biktima ondoko taulan adierazitako ordezkapen politiken arabera. Ondoko taula betetzea eskatzen da, bakoitzeko adieraziz ere zein orri-biktima aukeratuko litzatekeen eta aurreko taulako zein ezaugarrietan oinarritzen den ordezkapen algoritmo bakoitza azalduz:

<u>Ordezkapen algoritmoa</u>	<u>Zein ezaugarrietan</u> <u>oinarritzen den</u>	<u>Aukeratutako orri biktima</u>
FIFO		
2.aukera		
NRU		
NFU		
Zahartzearen algoritmoa		

b) Sistema orrikatu horretako orri taulen definizioak ondokoak izanik:

```
struct sarr_ot {
    BIT V;          /* OT baten sarrera bakoitza */
    BIT R;          /* baliagarritasun bita */
    BIT M;          /* erreferentzia bita */
    BIT K;          /* aldatuaren bita */
    int KargaTick; /* karga ticka */
    int ErreferentziBufferra; /* erreferentzien kontagailua,
                                bufferra, ala desplazamendu
                                erregistroa */
    int markoa;     /* marko zenbakia */
} OT [PROZ_KOP][OT_LUZERA]; /* prozesu guztien orri-taulak */
```

Implementa ezazu (C edo antzeko sasi-kodea erabiliz) ondoko orrien ordezkapen-algoritmoa:

```
int OrdezkapenAlgoritmoa (int p)
```

zeinak parametro bakarra duen (prozesuaren pid-a), eta FIFO politika eta asignazio lokala erabiltzen duen. Funtzio honek orri-biktimaren marko zenbakia itzultzen du.