



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

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

Estimatutako azterketaren iraupena: 2 ordu t'erdi

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

1 Ariketa [1,5 puntu]

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

1.a. [0,25 puntu] Sekzio kritikoaren arazoari aurre egiteko teknika orok lau baldintza beharrezko bete behar ditu. Etenen galarazpen/gaitzearen teknikak lau baldintzak betetzen al ditu? Arrazoitu zure erantzuna.

1.b. [0,25 puntu] Prozesuen planifikazioari dagokionez, aipa ezazu arrazoi bat lehentasun dinamikoen erabilera interesgarritzat jotzeko.

1.c. [0,25 puntu] Zein motako planifikazioa erabiliko zenuke Denbora errealeko sistema batean: *Round-Robin*, FCFS, kanporatzailea, ez-kanporatzailea,...?

1.d. [0,25 puntu] Azal ezazu gainorrikapena (*thrashing*) sor dezakeen arrazoiatariko bat, eta adieraz ezazu zein den zure proposamena arazo hau ekiditeko.

1.e. [0,25 puntu] Zer da *device driver* edo dispositiboaren maneatzailer bezala ezagutzen dena?

1.f. [0,25 puntu] Fitxategi-sistemei dagokionez, non gordetzen (edo kokatzen) da UNIXen kasuan fitxategien baimenen kontrol-informazioa? Eta FAT fitxategi sistemaren kasuan?

2 Ariketa [2 puntu]

Alegiazko memoriaren kudeaketa orrikatua jarraitzen duen sistema baten alegiazko helbideak 32 bitekoak dira eta orriak 4 Kbytekoak. Orri-taulako sarrera bakoitza 4 bytekoa da (30 bit marko zenbakia adierazteko eta gainontzekoak kontrol bitetarako). Sistema hau 512Mbytekoa RAM memoria duen ordenadore batean instalatu da. Zera eskatzen da:

2.a. Helbide logiko eta fisikoen eskemak.

2.b. Orri-taularen tamaina.

Aurreko sistema segmentatu baldin badugu orri-taulak marko bat baino handiagoak izan ez daitezten, ondoko galderak erantzun itzazu:

2.c. zer tamaina izan beharko du orri-taula bakoitzak?

2.d. Zenbat sarrera izango ditu segmentu taulak?

2.e. Zein izango da bit-maparen tamaina?

2.f. Prozesu baten segmentuen tamaina ondokoa da: 2Kbyte koderako, 0,5 Kbyte datuetarako, eta 1 Kbyte pilarako. Balio hauen ondorioz, zein izango da prozesu honengandik espero dezakegun barne-fragmentazioa sistema segmentatu-orrikatuan? Eta sistema orrikatu hutsean?

2.g. Eta zenbatekoa izango da kanpo-fragmentazioa azkeneko bi kasutan?

2.h. Working-setaren teknika erabiltzen badugu, kalkula ezazu sistemaren multiprogramazio-maila maximoa 512 Mbyteko RAM memoriaren kasurako, jakinik working-setaren tamaina maximoa prozesu bakoitzeko 8 orrikoa dela.

3 Ariketa [3 puntu]

UNIXeko fitxategi-sistema ezarri dugu ondoko ezaugarriak dituen disko baten gain: 1024 zilindro, 4 pista/zilindro, 64 sektore/pista eta 512 bytetako sektoreak. Jakina da ere diskoaren kontroladoreak sektore bat DMA bidez transferitzeko batezbesteko denbora 5 μ s-koa dela, diskoaren errotazio-abiadura 7.200 bira/minutukoa dela, eta buruaren kokapenerako batezbesteko denbora 9 ms-koa dela. Fitxategi-sistema honek Kbyte bateko blokeak erabiltzen ditu, eta bloke erakusleak 2 bytekoak dira. Fitxategi sistema honetan 3 fitxategi dauzkagu soilik, FitxA, FitxB eta FitxC, eta bakoitzari dagokion inodoen kokapen-informazioa ondoko irudian ikus daiteke:

FitxA-ren inodoa	atributuak	FitxB-ren inodoa	atributuak	FitxC-ren inodoa	atributuak
	8		11		15
	2		6		5
	3		9		null
	12		4		null
	null		7		null
	null		13		null
	null		10		null
	null		31		null
	null		23		null
	null		20		null
	null		24		null
	null		21		null
	null		null		null

a) Zein da FitxA eta FitxC fitxategien tamaina bloketan?

b) Zein da FitxB fitxategiak izan dezakeen tamaina minimo eta maximoa bloketan?

c) UNIXeko fitxategi sistema honetarako, kalkula ezazu disko honetan gorde daitekeen fitxategi handienaren tamaina (arbuigarri kontsidera ezazu inodoen kudeaketan galdutako disko-tartea).

- d) Sistema eragileak FitxA fitxategia atzitzen ari bada, eta uneko posizioa (karakteretan) 2457 bada; zein da helbideen itzulpenean erabiliko den prozedura (adierazi zeintzuk diren bloke erlatibo, bloke absolutu, eta zilindro-pista-sektore balioak? Adierazi ere pausu bakoitza S/I errutinak ala disko gestoreak burutzen duen.
- e) Zein da bloke físiko (sektore) bakar baten atzipen denbora?
- f) Aurreko ataletan lortutako emaitzez baliatuz, eta diskoak 2 faktoreko interleaving-a duela kontuan izanik, kalkula ezazu zein den bloke logiko oso bat atzitzeko denbora.

- g) Suposatuz diskoan dauden hiru fitxategi hauek katalogo nagusian daudela, eta diskoaren edukia blokez-bloke kopiatzen badugu ezaugarri fisiko berdineko beste disko batetara baina FAT fitxategi sistemarekin formateatuta dagoena, zein izango da disko berri honetan izango dugun kontrol informazioa katalogo nagusian eta FAT egituraren? Guzti hau adierazteko, bete itzazu ondoko taulak, eta FAT egituraren kasuan ez izan kontuan 15. bloketik aurrerakoak.

KATALOGOEA				FAT	
Izena		Has.	Luz.	Diskoaren Boot sektorea	
FitxA					0
FitxB			X		1
FitxC					2
					3
					4
					5
					6
					7
					8
					9
					10
					11
					12
					13
					14
					15
			

OHARRA: ariketa erraztearren, kontsidera ezazu FitxB fitxategian aurreko inodoen irudian adierazten ez diren blokeak 30 edo goragoko bloketan kokatzen direla.

- h) FAT disko berrian, fitxategi sistemaren eta diskoaren parametroak aurreko UNIX diskoaren berdinak direla kontsideratuz (hots, bloke tamaina berdinak eta 2 byteko erakusleak), kalkula ezazu disko honetan gorde daitekeen fitxategi handienaren tamaina.

4 Ariketa [2,5 puntu]

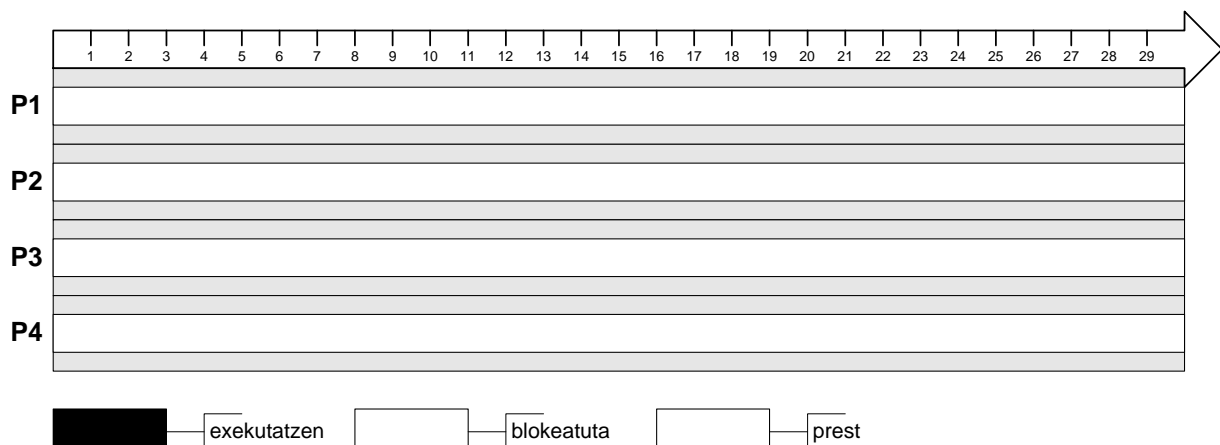
Sistema batean lau prozesu exekutatzen dira, ondoko taulan ematen delarik prozesu bakoitzari dagokion informazioa: identifikatzaileaz gain, bertan adierazten dira prozesu bakoitzaren sorrera tick-a, lehen eta bigarren CPU tarteen denbora, bi tarte hauen artean blokeatuta egongo den denbora (tick-etan), eta bakoitzaren lehentasuna (balio handiago batek lehentasun handiagoa adierazten du).

Prozesu guztiak antzeko jokaera erakusten dute: lehenengo CPU tarte igaro ondoren blokeatu egiten dira, ondoren bigarren CPU tarte exekutatuta eta azken hau betetzerakoan amaitu egiten dira (adib. P3-k 3 CPU tick exekutatuta ondoren, blokeatu egingo da 4 tick-etan zehar, eta beste 3 tick exekutatuta ondoren amaitu egiten da). Baldintza berdinen aurrean bi prozesu edo gehiago aukera badaitezke, *schedulerrak* identifikatzaile txikiena duena aukeratuko du.

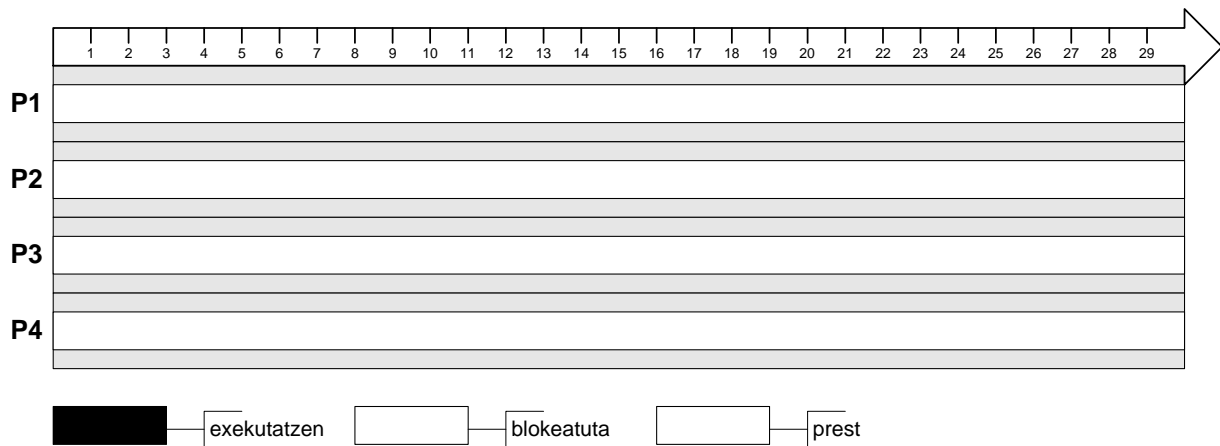
Prozesua	Sorrera tick	Lehenengo CPU tarte	Denbora blokeatuta	Bigarren CPU tarte	Lehentasuna
P1	0	7	1	4	1
P2	1	1	7	3	4
P3	2	3	4	3	3
P4	3	2	4	2	4

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.



- 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, $q=2$ 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						

5 Ariketa [1 puntu]

Sistema orrikatu baten orri-taulen definizioak ondokoak izanik:

```
struct sarr_ot {                /* OT baten sarrera bakoitza */
    BIT V;                      /* baliagarritasun bita */
    BIT R;                      /* erreferentzia bita */
    BIT M;                      /* aldatuaren bita */
    int KargaTick;             /* karga tick-a */
    int ErreferentziaTick;     /* azken erreferentziaren tick-a
                               */
    int markoa;               /* marko zenbakia */
} OT [PROZ_KOP][OT_LUZERA];    /* prozesu guztien orri-taulak */
```

Implementa ezazu LRU ordezkapen algoritmoaren hurbilpena burutzen duen ondoko funtzioa (C edo antzeko sasi-kodea erabiliz):

```
int LRU (int p)
```

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

LRUren hurbilpen honetan, orri bakoitza atzitzen dugunean azken erreferentziaren tick-a gordetzen da. Algoritmo honek biktima bezala aukeratuko du erreferentzia tick txikiena duen orria. Bi orri edo gehiagok izango balute erreferentzia tick txikiena, FIFO politika aplikatuko litzateke hauen artean.