

Izena:

1 Ariketa [2,5 puntu]

Erantzun labur eta zehatz ondorengo galderak, zure erantzunak arrazoituz:

1. Azaldu zergatik *working-set* kontzeptua aplikatuz gainorrikapena saihesten den.
2. Prozesadorearen errendimendua irizpide gisa hartuz, zein da CPUrako prozesuen planifikaziorik egokiena; FCFS ala Round Robin?
3. Diskoaren informazioa sektoreka banatuta badago, zer dela eta erabiltzen dute S/ Iko errutinek blokea helbideratze unitate bezala?
4. Windows sistemek badaukate fitxategi sistema trinkotzeko tresna bat (*defrag*). Hala ere, ezaguna denez, FATek ez du informazioa jarraian kokatzen, eta horregatik ezin da kanpo-fragmentazioa sortu. Zein da beraz, “defrag” tresnaren helburua?
5. UNIXen, fifo motako ilarak (buzoiak edo mailbox-ak) fitxategi bezala tratatzen dira. Horri esker, posible da fitxategi arruntekin erabiltzen diren sistema-dei berak erabiltzea (open, close, read, write). Hala ere, buzoiek eta fitxategi arruntek daukaten portaera ez da bera; prozesu batek mailbox-a irakurtzen duenean, informazio *kontsumitu* egiten da. Honi esker, posible da ekoizle-kontsumitzaileen eskema erabiltzea. Nola egiten du UNIXek bi motako funtzionamendu horiek (fifo ilaretakoak eta fitxategi arruntetakoak) bereizteko.
6. Disko baten geometria ondorengoa da: 1000 zilindro, 20 pista/ zilindro eta 1024 sektore/pista. Sektore baten tamaina 512 byte da. Disko honek partizio bakarra dauka,

disko osoaren luzerarekin eta partizio horrek daukan fitxategi sistemak 2.560.000 bloke du.

- a. Zein da diskoaren tamaina?
 - b. Eta fitxategi sistema honek izan beharko duen blokearen tamaina?
 - c. Zer dela eta ezin da fitxategi sistema hori FAT16 izan?
7. RAID sistema erabiltzeko arrazoi bat sistemaren eraginkortasuna hobetzea da. Zein da bestea?

2 Ariketa [0,5 puntu]

Ondorengo taulan aurkitu daiteke mezu trukerako primitiboen implementazioa, sistema monoprosadore batean. Ikusten denez, *jaso()* funtzioaren implementazioa bete gabe dago. Osatu implementazioa:

```
struct buzoi {
    struct semaf elementuak;
    struct semaf hutsuneak;
    mezu_mota mezu[MEZU_KOP_MAX];
} buffer[BUZOI_KOP_MAX];

/* Hasieraketa */
for (i=1; i< BUZOI_KOP_MAX; i++){          /* Buzoi bakoitzaren
                                           semaforoak hasieratu */
    hasieratu_semaforoa(buffer[i].hutsuneak, MEZU_KOP_MAX);
    hasieratu_semaforoa(buffer[i].elementuak, 0);
}

void bidali (int i, mezu_mota m)
{
    jaitsi(buffer[i].hutsuneak);
    galarazi();
    gorde_informazioa(i, m);
    gaitu();
    igo(buffer[i].elementuak);
}

void jaso (int i, mezu_mota m)
{
}

}
```

3 Ariketa [puntu 1]

Sistema eragile batek hiru prozesu exekutatu behar du aldi berean. Hiru prozesuen kodea jarraian dator:

A prozesua	B prozesua	C prozesua
<pre>lock(zerraila2); x2= x2+1; lock(zerraila1); unlock(zerraila2); x1= x1+1; unlock(zerraila1);</pre>	<pre>jaitsi(semaforol); lock(zerraila2); x2= x2-1; unlock(zerraila2); ... igo(semaforol);</pre>	<pre>lock(zerraila1); x1= x1-1; jaitsi(semaforol); ... igo(semaforol); unlock(zerraila1);</pre>

1. Asignazio grafo bat erabiliz, adierazi ezazu elkarblokeaketa sortzen den egoeraren bat, prozesu bakoitza zein agindutan geldituko litzatekeen erakutsiz.

2. *Zerraila1* $x1$ aldagaiaren erabileran atzipen eskusiboa implementatzeko erabiltzen da. Era berean, *zerraila2*-k $x2$ aldagaiarentzako funtzio bera betetzen du. Hau kontuan hartuta, aldatu prozesu baten kodea elkarblokeaketa gertatzeko aukera saihesteko. Komentatu zein den egindako aldaketa eta elkarblokeaketarako baldintzen arteko balizko erlazioa.

4 Ariketa [0,5 puntu]

Sistema multiprogramatu batean ondorengo planifikazioa inplementatzen da; denbora konpartitua eta gertaera bidezko jaurtiketak. Sistema honetan segundoko ematen diren batezbesteko testuinguru aldaketa kopurua neurtu da eta baita bakoitzaren arrazoia ere atera:

Testuinguru aldaketaren arrazoia	Batezbesteko testuinguru aldaketa kopurua
Prozesu berriak sortzeagatik	5
Prozesuak amaitzeagatik	5
Prozesuen blokeaketak direla eta	15
Prozesuen desblokeaketak	15
Quantum-a agortzea	10
Guztira	50

Testuinguru aldaketa egiteko batezbesteko denbora 20 mikrosegundokoa da. Kalkula ezazu planifikazio honek suposatzen duen denborazko eraginkortasun galera, erantzunak arrazoituz;

- a) Denbora konpartitua dela eta
- b) Gertaera bidezko kanporaketak direla eta

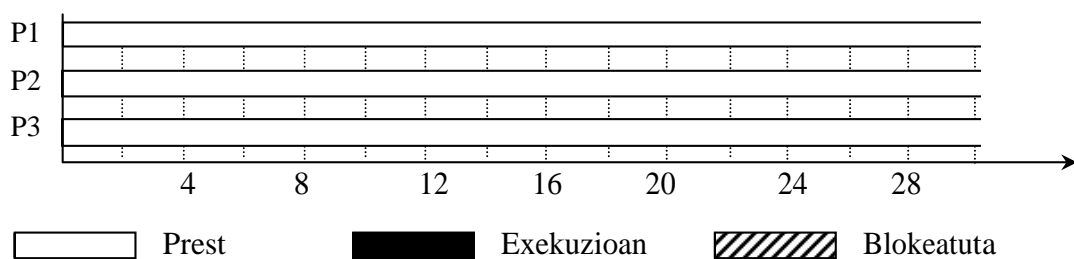
5 Ariketa [1,5 puntu]

Aurreko ariketaren sisteman, 4 tick-etako quantum-a konfiguratu egin da. Horrez gain, lehentasun dinamikoak erabiltzen ditu exekutatzera sartu behar duen prozesua aukeratzeko. Prozesuak, hasierako lehentasun batekin sortzen dira, eta quantum-a agortzen duten bakoitzean unitate batean gutxitzen zaie lehentasun hau 1 baliora iritsi arte. Suposa dezagun, instante berean hasierako lehentasuna 4 duten hiru prozesu martxan jartzen direla:

- P1, inoiz blokeatzen ez den kalkulura zuzendutako prozesua eta **orduetan** exekutatzen dena
- P2, S/I-ra bideratutako prozesua, zeinak txandaka, tick bateko tarteak exekutatu eta 5 tick-etan blokeatuta geratzen den, begizta hau errepikatuz etengabe.
- P3, 5 tick exekutatzen ditu, 4tan blokeatzen da, 4 exekutatzen ditu eta bukatu egiten da

Hasieran, P1, P2 eta P3 ordenan daude prest ilaran.

1. Prozesu hauen exekuzio kronograma marraztu (30. tick-a arte)



2. Kalkula ezazu P1, P2 eta P3ren CPU-tasak eta prozesuen batezbesteko latentzia

P1en CPU-tasa	P2ren CPU-tasa	P3ren CPU-tasa	Batezbesteko latentzia (bakarrik lehenengo aldian)

3. Planifikazio politika hau eta UNIXekoa alderatuz, zein antzekotasun eta diferentzia ikusten dituzu?

6 Ariketa [2 puntu]

32 bitetako alegiazko helbideak erabiltzen dituen sistema segmentatua diseinatu dugu, zeinak prozesuko 1024 segmentu onartzen dituen eta Gbyte bateko memoria fisikoa onartzen duen.

1. Alegiazko helbide eta helbide fisikoaren eskemak marraztu.
2. Segmentu-taula baten tamaina kalkulatu.
3. Komertzialki ez dira honelako sistemak inplementatzen. Hau horrela zergatik den azaltzen duten arrazoiak eman.

Izandako arrakasta txikia ikusita, alegiazko helbideratzea aldatzea erabakitzen da. Honetarako, segmentuak 8 Kbytetako orritan banatzen dira gainontzeko parametroak mantenduz. Sistema berri honentzako jarraian datozen galderak erantzun:

4. Alegiazko helbide eta helbide fisikoaren eskema berriak marraztu
5. Jarraian datorren taula bete, lehenengo zutabearen aipatzen diren elementuetatik zenbatek prozesu bat eta segmentu bat izan ditzaketen adieraziz (gehienez). Gelaxkaren batek zentzurik ez balu X bat jarri.

	Prozesu batek izan dezake ...	Segmentu batek izan dezake ...
Segmentu-taulak		
Orri-taulak		
Segmentuak		
Orriak		

6. Orri-taula bakoitzaren tamaina kalkulatu.

7. Libre dagoen espazioaren kudeaketa bit-maparen bidez egiten bada, onargarria den memoria fisiko guztia instalatua edukiz gero, kalkulatu zein den bere tamaina.

8. Kalkulatu 15 segmentu dituen 5Mbytetako prozesu baten barne-fragmentazioaren gutxi gorabeherako estimazioa.

9. Kalkulatu programa honek sortzen duen kanpo-fragmentazioaren estimazioa.

7 Ariketa [puntu 1]

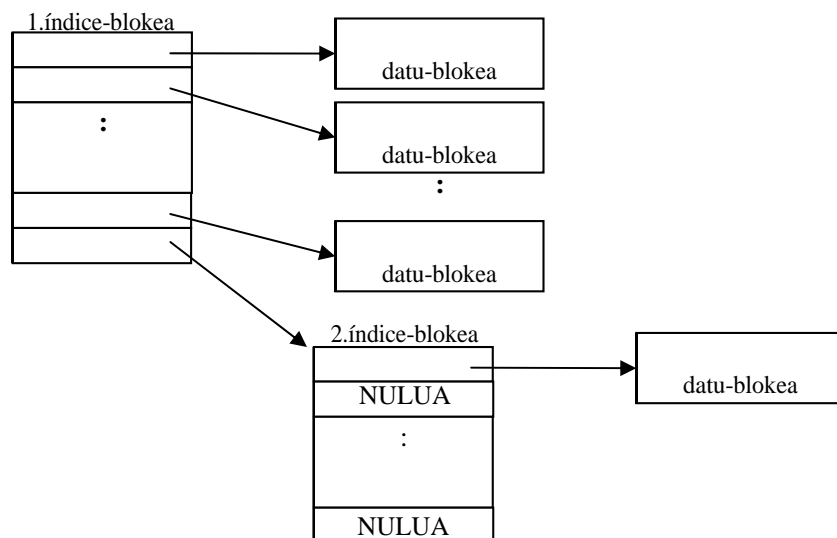
Alegiazko memoriadun sistema batek orrialdeen ordezkapenerako bigarren aukerako politika erabiltzen du, asignazio lokalarekin eta programa bakoitzeko gehienez 4 orriekin.

P programaren exekuzioak beheko taulako goiburuan espezifikatzen diren orrien erreferentzia-sekuentzia sortzen du. Kontutan hartu, hasieran 0, 4 eta 7 orrialdeak daudela memorian kargatuta eta denek beren erreferentzi bita 1era dutela. Memoriaren eboluzioa taulan deskriba ezazu orri-hutsegiteak noiz gertatzen diren adieraziz eta orri-biktima zein izan den esanez.

		Pren erreferentzia-sekuentzia															
		0	7	1	1	2	4	1	1	5	1	3	0	3	0	5	
Pren orriak memorian	0																
	4																
	7																
	--																
Biktima																	

8 Ariketa [puntu 1]

Kokapen indexatua erabiltzen duen fitxategi sistema batek indize-blokeak modu kateatuan kokatzen ditu.



Fitxategi baten lehenengo indize-blokea direktoriotik erakusten da.

Indize-bloke baten indize guztiek datu-blokeak erakutsiko dituzte, azkeneko indizeak izan ezik.

Azken honek, *NULUA* balioa ala hurrengo indize-blokea erakusteko erabiliko da, datu-blokeak adierazteko indize gehiago behar direnaren arabera.

Sistemak 32 bitetako erakusleak erabiltzen ditu eta blokearen tamaina Kbyte batetakoa da.

1. Direktorio bateko fitxategi hauek izanda beren tamainekin, adierazi kasu bakoitzean zenbat bloke esleitu behar diren fitxategi bakoitza adierazteko.

Izena	Tamaina	Índice-blokeak	Datu-blokeak	Guztira
FITX1	2 byte			
FITX2	400Kbyte			

2. Zein abantaila dauzka, fitxategien tamainari erreparatuz, kokaleku honek UNIX-ek erabiltzen duenaren aldean?
3. Zeinek beharko du atzipen gutxiago FITX2 fitxategi osoa irakurtzeko: sistema honek ala UNIXek