



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

*Estimatutako azterketaren iraupena: 2,5 ordu*

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

### 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.** Gure etxeko ordenadorearen CPUa 500 MHz-etakoa da eta erlojuaren arreta errutina 100 Hz-koa. Gure CPUa hobetzen badugu 1 GHz-ko CPU bat ezarriz, zertan aldatuko da sistemaren erlojuaren maiztasuna makina berrian? Arrazoitu zure erantzuna.

**1.b.** Zergatik erabili ohi da ostrukaren algoritmoa zenbait sistema eragiletan elkarblokeaketan arazoari aurre egiteko? Zertan datza?

**1.c.** 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...).

<b>Helbide-itzulpenaren unekak</b>	<b>Nork egiten duen</b>	<b>Exekuzio modua (erabiltzaile/sistema)</b>	<b>Erabilitako datu-egiturak</b>
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 schedulera...
- 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. Zer dira dispositiboen deskribatzaileak? Sistema Eragileko zein errutinek atzitzen dituzte?

1.e. Zertan datza NFU (Not Frequently Used) ordezkapen-politika? Honen hobekuntza suposatzen duen beste ordezkapen-politikaren bat ezagutzen al duzu? Zein da hobekuntzaren funtsa? Arrazoitu zure erantzuna.

## 2 Ariketa [2,5 puntu]

Sistema Eragile baten prest ilarara une berdinean hiru prozesu iritsi dira. P1 eta P2 prozesuek A programa exekutatzen dute, eta P3 prozesuak B programa. Bi programa hauek oso jokaera berdintsua dute: CPU eta blokeaketa tartekak dituzte, eta amaitu egiten dira, ondoko ezaugarrien arabera:

```

      A programa
{
    (2 CPU tick)
    (2 tick blokeatuta)
    (3 CPU tick)
    (2 tick blokeatuta)
    (1 CPU tick)
}

```

```

      B programa
{
    (2 CPU tick)
    (3 tick blokeatuta)
    (1 CPU tick)
}

```

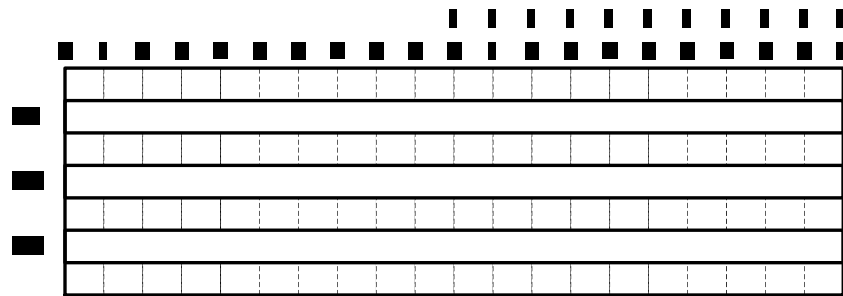
Honez gain, ondoko taulan prozesu bakoitzaren lehentasunak adierazten dira:

<u>Prozesua</u>	<u>Lehentasun-maila</u>
P1	1
P2	1
P3	3

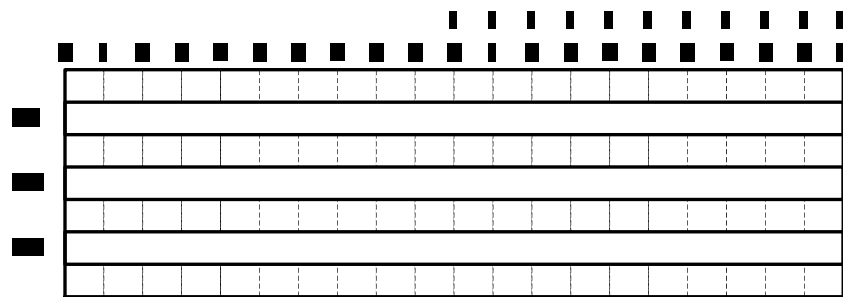
Kontuan izanik:

- lehentasun-maila handiagoak lehentasun handiagoa adierazten duela
- planifikazio politikaren arabera bi prozesu edo gehiago aukeratu badaitezke exekutatzeraz pasatzeko, beti aukeratuko dela identifikadore txikiena duena

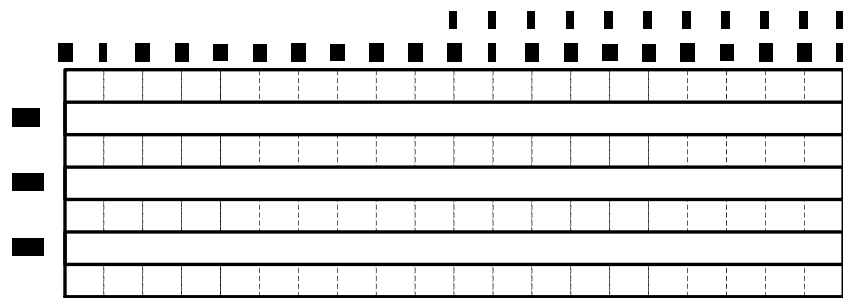
2.a. Marraz itzazu ondoko exekuzio-kronogramaren lehendabiziko 20 tick-ak, ondoko planifikazio politikarako: **FCFS eta ez-kanporatzailea gertaera nahiz denbora bidez.**



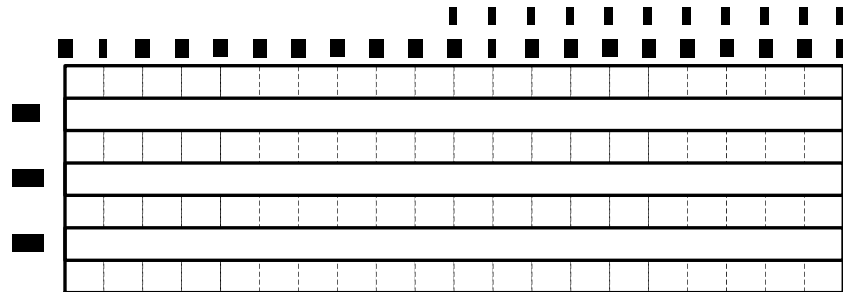
2.b. Halaber, marraz itzazu ondoko exekuzio-kronogramaren lehendabiziko 20 tick-ak ondoko planifikazio politikarako: **lehentasun estatikoa eta gertaera bidez kanporatzailea.**



2.c. Marraz itzazu ondoko exekuzio-kronogramaren lehendabiziko 20 tick-ak ondoko planifikazio politikarako: **FCFS eta Round Robin-ekin**, quantumaren balioa 1-ekoa izanik.



2.d. Marraz itzazu ondoko exekuzio-kronogramaren lehendabiziko 20 tick-ak ondoko planifikazio politikarako: **lehentasun estatikoa, gertaera bidez kanporatzailea, eta Round Robin-ekin**, quantumaren balioa 1-ekoa izanik.



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

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

### 3 Ariketa [2,5 puntu]

Memoria orrikatua duen memoria-sistema bat dugu, ondoko ezaugarriak dituena: 512 Mbyte RAM (nahiz sistemak onartzen duen 1Gbyte RAM helbideratu ahal izatea), 32 biteko helbide logikoak, eta 16 Kbyteko orriak. Memorian tamaina desberdineko 32 prozesu kargatu nahi ditugu, batez beste 512Kbyteko memoria esleituta dutelarik hauetako bakoitzak. Guzti hau kontuan izanik, erantzun itzazu ondorengo galderak, erantzunak arrazoitu behar direla kontutan hartuz:

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

**3.b.** Kalkulatu orri-aularen sarrera-kopurua

**3.c.** Zein da orri-aula bakoitzaren tamaina? Eta zenbat memoria-tarte esleitzen da prozesu guzti hauen orri-aulak gordetzeko memorian?

**3.d.** Zein da batez besteko kanpo-fragmentazioa prozesu bakoitzeko eta guztira?

**3.e.** 32 prozesu horiek kontuan izanik, gainorrikapen (*thrashing*) arazoak izango ditugula esan al daiteke? Baiezko kasuan, zein parametro handitu beharko genuke arazo hau konpontzeko?

Sistema honetatik abiatuz sistema berri bat ezarri nahi dugu alegiazko memoria segmentatuz, hala nola orri-taula bakoitza marko bateko tamainaren berdina izan dadin.

**3.f.** Marraz ezazu eskema berri honetarako alegiazko helbidearen eskema ?

**3.g.** Zenbat segmentu izan ditzake gehienez prozesu bakoitzak? Eta zenbat orri segmentu bakoitzeko?

**3.h.** Kalkula ezazu sistema eragileak memoria sistema berri honetan prozesu guztien segmentu-taulak memorian gordetzeko esleitu beharko duen memoria tartea.

**3.i.** Sistema berri honetan , aurreko 3.d atalean kalkulaturako kanpo-fragmentazioaren estimazioa aldatuko al da? Zenbatean? Arrazoitu zure erantzuna.

**3.j.** Eskema berri honetan exekutatu diren prozesuak desberdinak direla jakinik, non prozesu bakoitzaren working-setaren tamaina 2 segmentukoa izango den, zenbateko izango da multiprogramazio-maila maximoa?

#### 4 Ariketa [2 puntu]

Kontsidera ezazu ondoko kode-atala:

```
fp= open("fitx1", IRAKURRI);  
/* fich1 fitxategia irekitzen da irakurketa moduan */  
lseek(fp, 2050); /* irakurketa-erakuslea 2050.bytean kokatzen da */  
read(fp, &buff, 1100); /* 1100 karaktere irakurtzen dira */
```

zeina FAT16 erabiltzen duen sistema eragile baten gain exekutatzen den, non bloke logikoak 1024 bytekoak dira eta ondoko egoera duen:

KATALOGOA				FAT	
Izena	...	Luzera.(bytetan)	1 <sup>o</sup> blokea		
fitx1	...	4000	13	10	...
fitx2	...	205	11	11	EOF
fitx3	...	9200	15	12	7
...			...	13	14
				14	12
				15	LIBRE
				16	...

**4.a.** Fitxategiaren zein bloke erlatibo(et)an daude irakurritako 1100 karaktereak? Eta zein bloke absolutu(et)an?

**4.b.** Zein izango da fitx1 eta fitx2 fitxategien barne-fragmentazioa?

**4.c.** Zer gertatzen da fitx3-rekin? Zein izan daiteke egoera honen arrazoia?



**4.d.** Jakinda disko unitate honen errotazio-abiadura 600 bira/min-ekoa dela, disko unitatearen geometria  $(Z,P,S) = (80,4,8)$  dela, burua edozein zilindro batean kokatzeko batez beste 10mseg behar direla, bloke fisikoaren tamaina Kbyte batekoa dela, eta DMA bidez sektore bat transferitzeko denbora 3 mseg-koa dela, kalkula ezazu diskoaren driverrak 1100 karaktere horiek irakurtzeko beharko duen denbora. Halaber, kontsidera ezazu interleaving-ik ez dela aplikatu eta dispositiboaren bufferraren tamaina 2Kbytekoa dela.

**4.e.** FAT fitxategi-sistema erabili beharrean UNIX moduko erabili izan balitz parametro berdinak mantenduz (bloke tamaina eta kopuru berdinak), nola adieraziko litzateke fitx1 eta fitx2 fitxategien kokapenaren informazioa?

## 5 Ariketa [1,5 puntu]

FAT fitxategi-sistemaren ondoko definizioak emanik:

```
struct katalogo_sarrera {      /* katalogo sarrera bakoitzeko informazioa */
    ...
    char izena[8]
    int luzera;                /* bloketan */
    int lehenengo blokea;
    ...
} dir[FITX_KOPURU_MAX];

int fat[1024];                /* FAT taularen definizioa */
```

Sistema eragilearen ondoko funtzioa implementatuko dugu:

```
int BlokeAbsolutua (char *fitx_izena, int BlokeErlatiboa)
```

zeinak fitxategi baten izena emanda bere bloke erlatibo konkretu bat FAT disko unitatean zein bloke absolutuan kokatzen denaren informazioa itzuliko duen. Horretarako, kontuan izango dugu bloke logikoaren tamaina Kbyte batekoa dela.

**5.a.** Azal ezazu, hitzez edo modu eskematiko batean, zein den errutina honek jarraitu beharko duen prozedura.

**5.b.** Programa ezazu errutina hau C edo antzeko sasi-lengoai bat erabiliz.