Izena: *Taldea: euskara taldea*

Sistema Eragileen Diseinua 2002eko otsaila

1 Ariketa [1,5 puntu]

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

a) Aipa ezazu zein sistema eragile multiprozesadoreen berezitasunetariko bat monoprozesadorekoekin konparatuz.

b) Aipa ezazu denbora errealeko sistema batek bete behar dituen ezaugarrietariko bat. Arrazoitu zure erantzuna.

c) Zer dira dispositiboen deskribatzaileak? Sistema Eragileko zein funtziok atzitzen dituzte?

d) Prozesuen kudeaketari dagokionez, zein mekanismo ezagutzen dituzu multiprogramazio-maila mugatzeko?

e) Aipa itzazu gutxienez bi mekanismo alegiazko sistema batean sistemaren errendimendua hobetzeko.

f) Scandisk izeneko windows-eko aplikazioa fitxategi-sistemaren trinkotasuna (*konsistentzia*) aztertu eta konpontzeko erabiltzen da FAT fitxategi sisteman. Aipa ezazu FAT motako fitxategi sistema batean aurki daitekeen inkonsistentzia mota bat.

g) Zer dela eta ez da komenigarria sekzio kritikoen kontrola etenen galarazpenaren bitartez bideratzea multiprozesadoreetan?

h) Gure etxeko mikrokonputagailuen helbide busean zein nolako helbideak dituzu, alegiazkoak ala fisikoak? Zein hardware atalean egiten da helbide logikotik fisikorako itzulpena?

i) Zer da plug & play izenaz ezagutzen den ezaugarria?

2 ariketa [0,5 puntu]

Demagun disko jakin batean ondoko geometria dugula: 10 zilindro, 2 pista zilindroko eta 6 sektore pista bakoitzeko. Ondokoa eskatzen da: (a) Marraz ezazu eskema bat bloke fisikoak diskoan nola kokatzen diren adieraziz; bloke bakoitza zein disko zilindro, pista eta sektorean kokatzen diren argitu behar duzu (lehenengo bi edo hiru zilindroetako blokeen antolaketa marraztea baino ez da eskatzen). (b) Kalkula ezazu diskoaren gestoreak 4. bloke fisikoa atzitu nahi duenean erabiliko dituen disko-parametroen balioak (zilindro, pista, sektorea). Azken atal hau egiteko ez dira formulak erabili behar, marraztutako iruditik ondorioztatu behar dira parametro hauek.

3 ariketa [puntu 1]

Konputagailu baten fitxategi-sistemaren tartea helbideratzeko 32 biteko bloke erakusleak erabiltzen ditugu, eta fitxategi sistemaren blokearen tamaina (bloke logikoa) 32Kb-ekoa da. Diskoaren tamaina mugagabea dela kontsideratuz, kalkula ezazu FAT32n oinarritzen den fitxategi-sistema baten helbideratze tartea eta fitxategi baten tamaina maximoa.

Errepika itzazu egindako kalkuluak UNIX moduko fitxategi-sistema bat erabiltzen badugu. Jarraian aipatzen den taula erabil dezakezu emaitzak adierazteko. Taulan ematen dituzun balioen kalkuluak zeintzuk diren azal itzazu ezkerraldean.

	Helbideratze tartea	Fitxategiaren tamaina max.
FAT 32		
UNIX		

4 ariketa [1,5 puntu]

Ordenadore batek ondoko geometria duen disko gogor bat du instalatuta: 2000 zilindro, 8 pista/zilindro, eta 512 bytetako 400 sektore pista bakoitzeko. Diskoak 5400 bira/min-ko errotazio abiadura du eta irakurri/idazteko burua zilindro batean kokatzeko batez beste 10 mseg behar ditu. 15 µseg behar dira sektore bat (bloke fisiko bat) diskotik memoriara transferitzeko. Disko honetan swap partizio bat ezartzen da sistema eragile zahar baten alegiazko memoriaren kudeaketa egin ahal izateko. Orri-hutsegite bat gertatzen denean, orri biktima aldatua izan bada, orri-hutsegitearen errutinak une horretan bertan diskoan eguneratuko du orri bakar hori. Memoriako orriak 512 bytekoak dira.

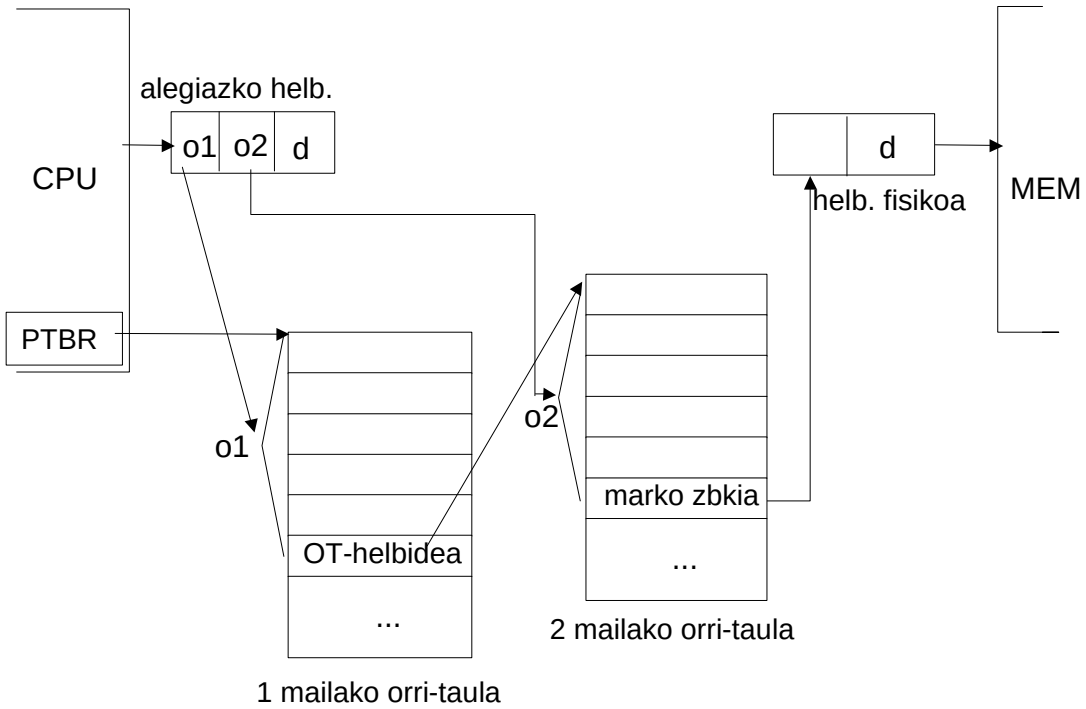
1. Kalkula ezazu orri bat diskoan idazteko denbora osoa.

2. Sistema eragileak igogailuaren algoritmoa (C-SCAN) erabiltzen du disko atzipenak kudeatzeko. Diskoaren burua 27. zilindroan kokatuta badago eta driverraren ilaran ondoko zilindroak atzitzeko ondoko 4 eskaera baldin badaude: 113, 328, 77 eta 113, (a) adieraz ezazu zein ordenan tratatuko diren eskaerak, eta (b) kalkula ezazu 4 eskaerak tratatzeko batezbesteko denbora (ez du garrantzirik atzipenak zein pista edo sektoreen gain egiten diren).

3. Gure etxeko ordenadorearen manualak irakurriz jakin dugu disko kontroladorea 4Kbtako RAM memoria bat duela. Ondorioz, kontroladoreak 8 sektore kontsekutiboen transferentzia denak batera une batez egiteko aukera eskaintzen du. Bestalde, irakasgaian ikasi dugunez badakigu Windows 2000ko alegiazko memoria sistemak aldatutako orriak taldeka eguneratzen dituela diskoan. Aldiz, instalatuta genuen sistema eragile zaharrak orriak banan bana eguneratzen ditu diskoan. Windows 2000 instalatzera ausartu baino lehenago pena merezi duela ziurtatu nahi dugu, eta beraz ondokoa eskatzen dizugu (a) kalkula ezazu memoria nagusitik 8 orri diskora 8 sektore jarraietara transferitzeko Windows 2000ak beharko duen denbora, eta (b) Azaldu ezazu zein den sistema zaharrarekiko errendimenduaren hobekuntza (ehunekotan) atzipen denborari dagokionez.

5 ariketa [1,5 puntu]

Windows NT sistema eragilearen memoriaren kudeaketa alegiazko memorian oinarritzen da, alegiazko helbideak 32 bitekoak izanik eta orriak 4Kbytekoak. Gainera, sistema konbinatu orrikatu-orrikatua erabiltzen du: orrikapena bi mailatan egiten da, eta horrela lehenengo mailako orri bakoitzak bigarren mailako 1024 orri izan ditzake gehienez. Helbide itzulpenaren kudeaketa ondoko irudiak agertzen duen moduan egiten da:



Windows NT Sistema eragilea memoria fisiko tamaina desberdinekin lan egiteko gai da. Horretarako marko-zenbakia adierazten duen orri-taulako sarrera bakoitzak 4 byteko luzera du, nahiz une konkretu batean memoria fisiko oso handia ez izan eta 4 byte horiek guztiz ez erabili. Guk 128Mbyte RAM dituen ordenadore batean instalatu berri dugu sistema eragile hau. Guzti hau kontutan izanik, kalkula ezazu:

(a) Prozesu bakoitzeko lehenengo eta bigarren mailako orri-taulen kopurua.

(b) Zein da lehenengo mailako eta bigarren mailako orri taulen tamaina bytetan?

(c) Sistema orrikatu hutsa erabili izan balitz, hau da, orrikapen maila bakar bat eta alegiazko helbidearen tamaina berdina izango balitz (32 bit), zein izango litzateke prozesu bakoitzaren orri-taularen tamaina bytetan?

(d) Memoria librearen kudeaketa bit-maparen bidez egingo balitz, zein izango litzateke bit-maparen tamaina 128Mbyteko ordenadore batean?

(e) Prozesu bat emanik, kode atalaren tamaina 1 Mb-koa bada, datuen atalaren tamaina 0,5 Mb ingurukoa bada, eta pila 1 Mb-koa bada (azken hau zehatz-mehatz), zein izango da prozesu honetan espero daitekeen barne-fragmentazioa? Kontsidera ezazu kode, datu, eta pila atalak lehenengo mailako orri-taulako sarrera independenteetatik helbideratzen direla.

(f) Eta zenbatekoa izango da aurreko (e) kasurako kanpo fragmentazioa?

(g) Sistema segmentatu-orrikatua izango balitz, zerbait aldatuko al litzateke aurreko bi ataletan? Arrazoitu zure erantzuna.

(h) Windows 2000 sistemak working-set teknika erabiltzen du. Kalkula ezazu 128Mb-eko memora fisikoa duen sistema baten multiprogramazio-maila maximoa prozesu bakoitzeko working-set tamaina minimoa 4 orrikoa bada.

6 ariketa [2,5 puntu]

Ondoren lau programa desberdinen kodea aurkezten da. Bertan agertzen dira ere atal bakoitzak eskatzen duen CPU denbora.

```
prog1(void)
{
   while (1) {
     jaitsi(sem_b1);
     erabili(b1);  --   (7 tick)
     igo(sem_b1);
     ...           --   (3 tick)
   }
}
```

```
prog2(void)
{
   while (1) {
     jaitsi(sem_b2);
     erabili(b2);  --   (5 tick)
     igo(sem_b2);
     ...           --   (2 tick)
   }
}
```

```
prog3(void)
{
     ...           --   (10 tick)
}
```

```
prog4(void)
{
   while (1) {
     jaitsi(sem_b2);
     erabili(b2);  --   (4 tick)
     igo(sem_b2);
     ...           --   (4 tick)
   }
}
```

Horrela, *prog1* programa exekutatzen duen prozesu baten kasuan adibidez, *b1* baliabidea eskatu ondoren 7 tick-etako sekzio kritiko bat exekutatzen du, *b1* atzipen esklusiboz duelarik, eta *b1* baliabidea askatu ondoren 3 tick exekutatuko ditu berriro ere *b1* eskatu baino lehen hurrengo bigiztan. Bestalde, *prog3* programaren kasuan, 10 CPU tick exekutatu besterik ez du egiten (ez da sekula blokeatzen) eta ondoren amaitu egiten da.

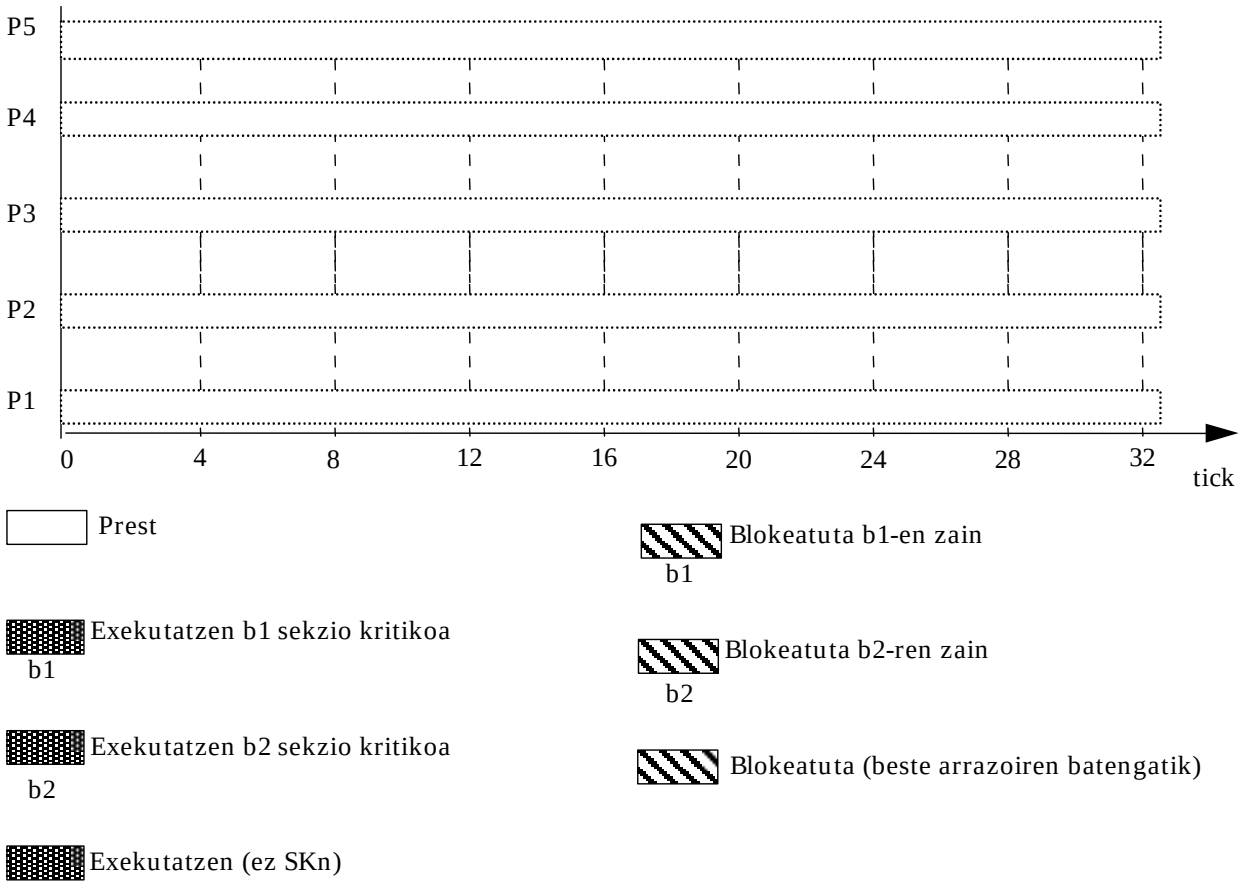
Semaforoen *jaitsi* eta *igo* eragiketak exekutatzeko denbora arbuiagarri bezala kontsidera itzazu.

Ondoko taulan sistema eragile batean exekutatzen ari diren 5 prozesuen izaera azaltzen da. Guztiak erabiltzaile mailako prozesu arruntak dira, eta taulan agertzen da bakoitzak exekutatzen duen programa eta erabiltzaileak emandako lehentasunak zeintzuk diren:

Proz.	*Exekutatzen duen programa*	*Lehentasuna*
P1	prog1	6
P2	prog1	4
P3	prog2	6
P4	prog4	6
P5	prog3	6

1) Elkar-blokeaketa arazorik gerta al daiteke programa hauek aldi berean exekutatzen direnean? Arrazoitu zure erantzuna.

2) Marraz ezazu prozesu hauen exekuzioaren lehenengo 32 tick-ak ondoko planifikazio politikarako: FCFS eta denbora konpartitua q=3 izanik. Horretarako kontsidera ezazu prozesu guztiak hasieran (0.tick-ean) aldi berean sortu direla, eta semaforo guztiak 'mutex' motakoak direla eta behar bezala hasieratuta daudela. Bi aukera desberdin egonez gero CPUra sartu behar den hurrengo prozesua aukeratzeko, zure irizpidea zein izan den azal ezazu.



3) Kalkula ezazu zein den sistema osoaren CPUaren erabilpen-tasa.

4) Kalkula ezazu zein den sistema osoaren batez besteko erantzun denbora (t_r).

5) Kalkula ezazu eraginkortasunaren galera (denbora portzentaian) sistema honetan denbora konpartitua dela eta, kontuan izanik tick bat 10 mseg-koa dela eta t_{cs} = 1 mseg (testuinguru aldaketa bakoitzeko denbora da t_{cs}).

6) Enuntziatuko programa guztietan bi sekzio kritiko desberdin bereizturik agertzen dira, semaforo desberdinak erabiliz *b1* eta *b2* baliabideak atzitu nahi ditugunean. Bi sekzio kritiko mota desberdin bereiztu beharrean baliabide guztiak atzitu ahal izateko semaforo bakar bat erabiliko bagenu, zein aldaketa aurreikus ditzakezu prozesuen CPU-tasan (ez da aldaketaren zenbatekoa kalkulatu behar, hobetu egiten den ala ez adierazi behar duzu)? Arrazoitu zure erantzuna.

7 **Ariketa** [1,5 puntu]

Sistema orrikatu batean orri taulen definizioak ondokoak dira:

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

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

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

zeinak parametro bakarra duen (prozesuaren pid-a), eta NRU politika eta asignazio lokala erabiltzen duen. Funtzio honek orri-biktimaren marko zenbakia itzultzen du. Orri-biktima aukeratzerako orduan, orrien arteko berdintasuna jarraitzen saiatu behar da algoritmoa.

7 Ariketa (2. aukera) [1,5 puntu]

Konputagailu batean bi CD unitate instalatu ditugu (DRIVE_D eta DRIVE_E bezala ezagutuko dira), idazketa eragiketarik onartzen ez dutenak. Tamalez, konputagailuaren sistema eragileak ez du driverrik CD unitateak atzitu ahal izateko, eta driver baten inplementazioa burutu nahi da. Sistema eragileak bi unitate hauen irakurketa egiteko bezero-zerbitzari eskema jarraituko du, beste disko unitateekin egiten den moduan. Edozein disko unitateren eskaeren formatua ondokoa da:

```
struct eskaera {
  struct item elementu_ilara;
  int pid;
  int unitatea;    /* DRIVE_A, DRIVE_C */
  int blokea;
  int eragiketa;   /* IRAKURKETA, IDAZKETA */
  char **pbuff;
  int gertaera;
  int errorea;
} IORB[ESK_KOP_MAX];
```

Bezeroek CD unitate bat atzitu ahal izateko ondoko funtzioari egingo diote dei:

```
/* CDen atzipena eskatzeko funtzioa: */

int CD_eskaera (int unitatea, int blokea, char **pbuff, int gert)
{
  int id;
  struct eskaera *esk;

  id= nor_naiz_nuk();
  esk= hartu_elementua_eskaera(id);
  esk->pid= id;
  esk->unitatea= unitatea;
  esk->blokea= blokea;
  esk->eragiketa= IRAKURKETA;
  esk->pbuff= pbuff;
  esk->gertaera= gert;
  esk->errorea= 0;
  ilaratu(&CD_gestore_ilara, esk);
  signal_nuk(GERTAERA_CD_GESTOREA);
  wait_nuk(gert);
  return(esk->errorea);
}
```

Driverraren eraikuntzan parte har dezazun eskatzen zaizu. Zure lana CD driverraren kudeatzailea programatzean datza. Ondoko informazioa izan behar duzu kontuan:

1. CD unitate guztien ezaugarri jakina da alde bakarrekoak direla.
2. Sistema eragileen nukleoan aurretik programatutako ondoko funtzioak existitzen dira edozein disko unitatea atzitu ahal izateko:

```
/* Nukleoko primitiboak edozein disko unitaterako: */

void motor_on_nuk (int unitatea)
void motor_off_nuk ()
int kokatu_pista_nuk (int unitatea, int pista)
int irakurri_sektorea_nuk ( int unitatea, int alde, int sektorea,
                            char *buff)
int idatzi_sektorea_nuk ( int unitatea, int alde, int sektorea,
                          char *buff)
int birkalibratu_nuk (int unitatea)
int wait_nuk (int semaforoa)
int signal_nuk (int semaforoa)
int nor_naiz_nuk ()
```

3. Aurredefinitutako hurrengo aldagaiak existitzen dira eta sistema eragile osoarekiko dira globalak. Berriak behar izanez gero definitu egin ditzakezu ere:

```
struct ilara CD_gestore_ilara;
int uneko_pista_A, uneko_pista_C;
int disko_egoera_A, disko_egoera_C;
                          /* GELDIRIK, ABIADURA_HARTZEN, MARTXAN */
```

Kudeatzaileak ondoko funtzioak bete behar ditu

- motorra martxan jarri geldirik badago
- erabiltzaileari CD unitatearen kokapen errore bat luzatu baino lehen bi saiakera egingo dira CDaren burua kokatzeko, eta bost saiakera egingo dira sektore bat irakurtzeko errore bat gertatu dela adierazi baino lehen. -1 errore-kodea itzuliko da buruaren kokapen errorea adierazteko, eta –2 paritate erroreetarako.
- SCAN eskaeren kudeaketa jarraituko da CD unitateetan, horretarako ondoko atzipen-funtzioa erabiliz:
  ```
  struct eskaera *scan(struct ilara eskaera_ilara, int unitatea);
  ```
- Bloke zenbaki bat emanik, bere pista eta sektore balioak lortu ahal izango ditu ondoko funtzioaren bidez:
  ```
  void formula_CD (int blokea, int *pista, int *sektorea);
  ```

OHARRA: kudeatzailea egiteko behar dituzun konstante, aldagai global eta funtzio berri guztien azalpen laburra eman behar duzu.