

# Sistemes Operatius II

## Avaluació parcial – Parcial part 2 – 19 de desembre del 2019

Nom i Cognoms: \_\_\_\_\_

En total hi ha 3 problemes. Podeu fer servir apunts o portàtil per fer consultes a les transparències però no es pot programar cap codi. Respondre breument a les preguntes amb un màxim de 4 pàgines (2 fulls). No es puntuaran preguntes no raonades.

**Problema 1 (concurrència, dificultat baixa, 2.5 punts).** Supposeu el següent codi

```
01  variables globals:
02      pthread_mutex_t mutex1, mutex2;
03
04  funcio1:
05      lock(&mutex1);
06      /* funcio 1, part 1 */
07      lock(&mutex2);
08      /* funcio 1, part 2 */
09      unlock(&mutex2);
10      /* funcio 1, part 3 */
11      unlock(&mutex1);
12
13  funcio2:
14      lock(&mutex2);
15      /* funcio 2, part 1 */
16      lock(&mutex1);
17      /* funcio 2, part 2 */
18      unlock(&mutex1);
19      /* funcio 2, part 3 */
20      unlock(&mutex2);
```

Es demana

1. En aquest codi les variables mutex1 i mutex2 són variables globals. Fa falta que siguin globals? No poden ser locals a cadascuna de les dues funcions, funcio1 i funcio2? Raona la resposta.
2. Indica quin problema es pot produir en aquest codi i per què. Raona la resposta indicant els canvis de context que s'han de produir perquè es produeixi el problema.
3. Comenta (breument) com solucionar el problema que es dona al punt 2. Indica com ha de ser el codi de funcio1 i funcio2 perquè el codi funcioni correctament.

**Problema 2 (semàfors, dificultat mitjana, 3.5 punts).** Un refugi de muntanya té dutxes unisex. Pot ser utilitzat tant per homes i dones però no hi pot haver homes i dones al mateix temps a la dutxa. Es proposa a continuació una solució al problema fent servir semàfors.

1. Per a què serveixen cadascun dels semàfors mutexD, mutexH i dutxa? Comenteu la vostra resposta per a cadascun dels semàfors en el context d'aquest codi.
2. Comenteu la solució proposada. Soluciona el codi el problema de l'exclusió mútua? És a dir, hi pot haver dones i homes al mateix temps a la dutxa? Supposeu que no hi ha límit en el nombre de persones que hi pugui haver a la dutxa. Comenteu la resposta.
3. Té algun defecte aquesta solució? Comenteu la vostra resposta.
4. (Difícil) Supposeu que la dutxa té una capacitat màxima per a quatre persones. Comenteu i indiqueu com modificaríeu el codi perquè a tot estirar hi pugui haver quatre persones (del mateix sexe!) a la dutxa. Podeu indicar les modificacions en el codi que hi ha sota i comenteu per què soluciona aquesta restricció. Assegureu-vos que no es produeix un deadlock.

```
01 variables globals:
02   int dones = 0, homes = 0;
03   sem_t mutexD (= 1), mutexH (= 1), dutxa (= 1);
04
05 dones:
06   sem_wait(&mutexD);
07   dones = dones + 1;
08   if (dones == 1) sem_wait(&dutxa);
09   sem_post(&mutexD);
10   // dutxar-se
11   sem_wait(&mutexD);
12   dones = dones - 1;
13   if (dones == 0) sem_post(&dutxa);
14   sem_post(&mutexD);
15
16 homes:
17   sem_wait(&mutexH);
18   homes = homes + 1;
19   if (homes == 1) sem_wait(&dutxa);
20   sem_post(&mutexH);
21   // dutxar-se
22   sem_wait(&mutexH);
23   homes = homes - 1;
24   if (homes == 0) sem_post(&dutxa);
25   sem_post(&mutexH);
```

**Problema 3 (monitors, dificultat mitjana, 4 punts).** El problema de "la barberia" és un dels problemes clàssics de la sincronització de fils. Es presenta a continuació una versió simplificada del problema original. La barberia és una botiga en què hi ha una sala d'espera amb infinites cadires, i la sala del barber on el client és atès. Les condicions de funcionament de la barberia són les següents: a) Només hi ha un barber. Si no hi ha cap client, el barber se'n va a dormir (línies 12-13 del codi), b) Quan una persona es vol tallar el cabell crida a la funció client. Si el barber està ocupat, el client espera (línies 23-24), c) Se suposa que el barber arriba abans a la botiga que els clients.

```
01 variables globals:
02     int clients = 0;           // Nombre de clients
03     boolean ocupat = false;    // Permet saber si el barber esta ocupat
04
05     mutex barberia;
06     cond cond_barber, cond_clients;
07
08 barber: // Hi ha nomes un unic barber. El barber obre la botiga.
09     while (true) {
10         lock(barberia)
11         ocupat = false
12         if (clients == 0) // Si no hi ha clients, el barber s'adorm
13             wait(cond_barber, barberia)
14         signal(cond_clients)
15         clients--
16         unlock(barberia)
17         tallar_cabell_client()
18     }
19
20 client: // Els clients criden la funcio quan es volen tallar el cabell
21     lock(barberia)
22     clients++
23     while (ocupat) // Si el barber esta ocupat, el client s'adorm
24         wait(cond_clients, barberia)
25     signal(cond_barber)
26     ocupat = true
27     unlock(barberia)
28     em_poso_a_la_cadira()
29     return
```

Comenteu i raoneu les respostes a les preguntes en el context d'aquest codi:

1. Fa falta protegir amb la mateixa clau "barberia" tant la part del barber com la del client? Es poden protegir aquestes dues parts amb claus diferents?
2. Observar que a la línia 12 es protegeix el "wait" amb un "if" en comptes d'un "while". Raoneu per què no fa falta un "while".
3. De forma similar, observar que a la línia 23 es protegeix el "wait" amb un "while" en comptes d'un "if". Raonar per què no es pot posar un "if". Es pot produir algun problema?
4. Comenteu el funcionament el codi amb un exemple. En particular, indiqueu quan el barber/client s'adormen i on es fan despertar. El barber obre la botiga i ho hi ha cap client; un client A entra a la botiga; mentre el client A és atès els clients B i C entren a la botiga; el barber acaba d'atendre el client A i passa a atendre els clients que queden a la botiga. Supposeu que mentrestant no arriben nous clients.
5. (Difícil) Supposeu que a la botiga hi ha múltiples barbers en comptes d'un únic barber. Com modificariéu el codi per tal d'adaptar-ho a aquest cas? A la pàgina següent se us torna a mostrar el codi que teniu en aquest full. Utilitzeu-lo per indicar les modificacions que hi faríeu. Comenteu també la vostra proposta.

Utilitzeu aquest codi per respondre al punt 5 del problema.

```
01 variables globals:
02     int clients = 0;           // Nombre de clients
03     boolean ocupat = false;    // Permet saber si el barber esta ocupat
04
05     mutex barberia;
06     cond cond_barber, cond_clients;
07
08 barber: // Hi ha nomes un unic barber. El barber obre la botiga.
09     while (true) {
10         lock(barberia)
11         ocupat = false
12         if (clients == 0) // Si no hi ha clients, el barber s'adorm
13             wait(cond_barber, barberia)
14         signal(cond_clients)
15         clients--
16         unlock(barberia)
17         tallar_cabell_client()
18     }
19
20 client: // Els clients criden la funcio quan es volen tallar el cabell
21     lock(barberia)
22     clients++
23     while (ocupat) // Si el barber esta ocupat, el client s'adorm
24         wait(cond_clients, barberia)
25     signal(cond_barber)
26     ocupat = true
27     unlock(barberia)
28     em_poso_a_la_cadira()
29     return
```