

Problema Conceptes Bàsics II

Amb els exercicis d'avui, es busca que us familiaritzeu amb com solucionar problemes en els que apareixen problemes de concurrència fent servir els mecanismes disponibles (semàfors i mutexs).

1. Dos pobles veïns es troben separats per un riu molt cabalós que només pot ser creuat fent servir un pont romà. Per aquest pont només poden creuar fins a 20 persones alhora en el mateix sentit. Això vol dir que:
 - Si una persona vol creuar en un sentit i hi han persones creuant en l'altre, caldrà que esperi fins que el pont quedi lliure.
 - Si una persona vol creuar en un sentit i davant seu hi han 20 persones creuant en el mateix sentit, caldrà que esperi a que el nombre de persones creuant sigui menor que 20 per no sobrecarregar el pont.L'acció de creuar el pont està modelada per una rutina (que no cal implementar) que anomenarem Creuar().

Escriviu el pseudo-codi que descriu el protocol que hauria de fer servir una persona que vol creuar d'un poble a l'altra de forma segura fent servir semàfors (com el protocol és simètric per creuar de qualsevol dels dos pobles a l'altre, tenim prou amb descriure un sol procediment). L'estat del sistema estaria descrit per dues variables compartides anomenades creuant_p1_p2 i creuant_p2_p1. Cada una d'aquestes variables indica el nombre de persones que en un moment determinat estan passant el pont del poble 1 al 2 o del 2 a l'1 respectivament.

La solució incompleta al problema seria:

```
int creuant_p1_p2 = 0, creuant_p2_p1 = 0;
<declaració i inicialització dels semàfors i mutexs>

Persona_p1_p2(){
  <protocol per decidir si pot creuar: (1) no ve ningú en sentit
  contrari i (2) no hi han més de 20 persones passant>
  Creuar();
  <protocol a seguir després de creuar el pont>
}
```

Completeu aquesta solució indicant clarament quins semàfors i mutexs fareu servir, el seu valor inicial, el codi que s'ha d'executar abans i després de creuar el pont.

2. Considereu el següent escenari. Una ciutat té un restaurant molt popular. El restaurant té capacitat per a N comensals. El nombre de persones de la població que desitgen menjar al restaurant, i que esperen fora de les seves portes, és molt més gran que N.

El restaurant gestiona el seu servei de la següent manera:

- Sempre que està llest per al servei, **obre la porta d'entrada** i espera que entrin els comensals.
- Un cop entren **N comensals**, **tanca la porta principal** i passa a **servir** aquests comensals.
- Un cop **finalitzat el servei**, s'obre la **porta del darrere** i es deixa **sortir** els comensals per la porta del darrere.
- Un cop han sortit tots els comensals, tornem al pas 1.

Modelem els comensals i el propi restaurant com a fils en un programa multifils. Els fils s'han de sincronitzar de la següent manera:

- Un comensal no pot entrar fins que el restaurant hagi obert la porta principal per deixar entrar la gent.
- El restaurant no pot començar el servei fins que no hagin entrat N comensals.
- Els comensals no poden sortir fins que la porta del darrere estigui oberta.
- El restaurant no pot tancar la porta del darrere i preparar-se per al següent lot de comensals fins que no hagin marxat tots els comensals del lot anterior.

A continuació es proporciona un pseudocodi (sense les sincronitzacions) per als fils del comensal i completat pel fil del restaurant. La vostra tasca és completar el codi del fil del comensal de manera que la simulació funcioni correctament. Se us donen les variables següents (semàfors i valors inicials, mutex i variables compartides) per utilitzar en la vostra solució. Els noms de les variables us han de donar una pista sobre el seu possible ús. No heu d'utilitzar cap altra variable a la vostra solució:

```
Tsem comensals_entrants=0, comensals_sortint=0,  
Tmux entrada_feta=0, sortida_feta=0, mux_entro=1, mux_surto=1;  
int comptador_entrada=0, comptador_sortida=0;
```

Tots els canvis als comptadors, semàfors i mutex els heu de fer vosaltres a la vostra solució. Cap de les accions realitzades pel codi donat a continuació modificarà cap de les variables anteriors.

```

Pseudocodi restaurant(){
    while(1){
        obrePortaEntrada()
        signal(comensals_entrants, 30) //Suma 30 al semàfor
        lock(entrada_feta)           //Esperem que entrin els comensals

        tancaPortaEntrada()
        ferServei()
        obrePortaSortida()
        signal(comensals_sortint, 30) //Suma 30 al semàfor
        lock(sortida_feta)           //Esperem que sortin els comensals
        tancaPortaSortida()
    }
}

```

```

Pseudocodi comensal(){
    // Entrada al restaurant (agafo plaça si puc, si no, m'espero)
    entraRestaurant()
    // Per poder menjar, cal que haguem entrat 30 comensals (algú ha
    d'avisar que ja hi som tots)
    menjar()
    // Deixo la meva plaça i me'n vaig
    surtRestaurant()
    // Per tancar el torn, cal que haguem sortit els 30 comensals (algú ha
    d'avisar que ja hem marxat tots)
}

```