

Exercice 1 : Communication interprocessus : Modèle producteur-consommateur

Soient deux classes de processus appelés Producteurs et Consommateurs, qui se communiquent de l'information (des messages) à travers une zone de mémoire commune (buffer). Les messages transmis au consommateur par le producteur (pour simplifier, on va considérer, sauf indication contraire, un couple producteur-consommateur) ont par hypothèse une taille constante, et le buffer est doté d'une capacité fixe de N messages ($N > 0$).

Les vitesses d'exécution des deux processus sont quelconques.

La communication entre processus doit respecter les règles suivantes:

- Le consommateur ne peut prélever un message en cours de dépôt par le producteur.
- Le producteur ne peut déposer un message dans le buffer lorsque ce dernier est plein; si tel est le cas, il doit attendre.
- Le consommateur doit prélever tout message une et une seule fois.
- Si le producteur (respectivement le consommateur) est en attente parce que le buffer est plein (respectivement vide) il doit être réveillé dès que cette condition devient fausse.

Les règles qui ont été imposées aux processus de production et de consommation régissent le bon fonctionnement de ce système en particulier:

- Assurer la cohérence et l'intégrité des messages déposés dans le buffer,
- Eviter la perte de messages, par effacement, dans le buffer,
- Prélèvement redondant d'un même message.

Questions: Ecrire les algorithmes des processus producteur et consommateur en utilisant les sémaphores, Pour m ($m > 1$) producteurs et p ($p > 1$) consommateurs avec un buffer de capacité $N > 1$

Exercice 2 : Problème des philosophes (E.W. Dijkstra)

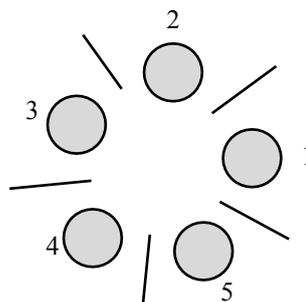
Cinq personnes assis autour d'une table prennent part à une curieuse habitude dont l'activité consiste (pour raison académique) à alterner entre penser et manger. Leur existence se résume selon le schéma suivant.

cycle

penser;
manger;

endcycle

Devant chaque personne se trouve un bol de riz (supposé toujours non vide). Quand une personne a faim, elle décide de manger et saisit les deux baguettes disposées de part et d'autre de son bol, voir figure ci-après (il n'y a que 5 baguettes sur la table). Une personne ne peut manger que lorsqu'aucun de ses voisins n'est entrain de manger.



Question : Ecrire l'algorithme modélisant l'activité d'une personne i ($1 \leq i \leq 5$) en utilisant :

- 1) Les sémaphores
- 2) les régions critiques;

Exercice 3 : Modèle de l'allocateur de ressources.

Soit un allocateur gérant un pool de N ressources identiques et constitué des deux procédures Allouer(n) et Libérer(n) dont les programmes ci-après sont sensés réaliser les opérations désirées.

mutex : *semaphore* *init* 1;
attente : *semaphore* *init* 0;
nlibre : *integer* *init* N;

```

natt : integer    init 0;
Procedure Allouer(n);
begin
  Demande: P(mutex);
  if n > nlibre then
    begin
      natt:=natt + 1;
      P(attente);
      aller_a Demande;
    end
  else
    nlibre:=nlibre - n;
  endif
  V(mutex);
end

Procedure liberer(n);
begin
  P(mutex);
  nlibre:=nlibre + n;
  while natt > 0 do
    begin
      natt := natt - 1;
      V(attente)
    end
  endwhile
  V(mutex);
end

```

Question 1

Les programmes précédents sont incorrects. Dites pourquoi, et donner les modifications nécessaires pour les rendre corrects.

Donner une solution en utilisant les régions critiques.