USTHB
Faculté d'Electronique et Informatique
Département Informatique

SERIE N° 2

Le 12/10/2021 - Année 2021/2022

Exercice 1 : Le programme suivant (algorithme de Dekker) est proposé comme solution au problème de la section critique entre deux processus concurrents :

Initialement : Di := faux ; (i= 1,2) tour := 1 ou 2 ;

Processus Pi
Début
Répéter
    Di := vrai :
    Tant que Dj faire si tour = j Alors
        Début Di := faux ; Tant que tour = j Faire <rien>Fait ; Di :=vrai Fin ;
    Fait ;
    <section critique i>
    tour : =j ; Di : = faux ;
Jusqu'à faux ;
Fin.

- Vérifier les conditions de la section critique.

Exercice2: Le programme suivant (algorithme de Lamport) est proposé comme solution au problème de section critique entre deux processus concurrents ;

Choix : tableau [0.. 1] de booleen :=faux; Numero : tableau [0.. 1] de entier :=0 ;

Notation : $(a,b) < (c,d) \Leftrightarrow (a < c)$ ou $(a = c$ et $b < d)$

Processus Pi
Debut
Répeter
$Choix_i$= vrai ; $Numero_i$= max ($Numero_i$, $Numero_j$) +1 ; $Choix_i$= Faux ;
Tant que $choix_j$ faire rien fait ;
Tant que ($Numero_j \neq 0$) et (($Numero_j$, j) < ($Numero_i$, i)) faire <rien> fait;
<SC_i >
$Numero_i$ = 0 ;
<Section $restante_i$ > ;
Jusqu'à faux ;
Fin ;

a/ Vérifier les conditions de la section critique
b/ Discuter le problème lié à la valeur numéro qui peut croître indéfiniment
c/ Réécrire la solution pour n processus

Exercice 3 :

On propose la solution suivante comme protocole d'exclusion mutuelle pour deux processus Pi et Pj.

Initialement Di=Dj=faux ; tour= i ou j ;
Processus Pi
Début
Répéter
    Di := vrai :
    Tantque tour = j **Faire**
        Tantque Dj = vrai **Faire** <rien>**Fait** ; tour :=i
    Fait ;

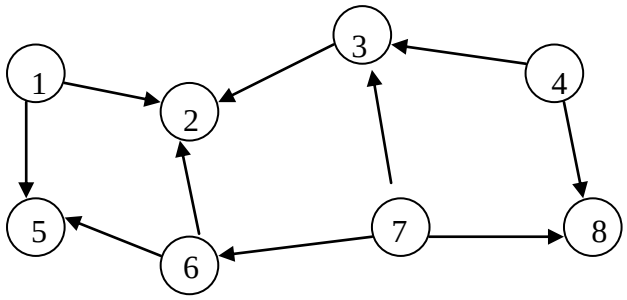
<Section Critique i>
Di : = faux ;
Jusqu'à faux
Fin.

1/ Donner la condition d'entrée en SC.
2/ Vérifier si cette solution peut être retenue comme protocole d'exclusion mutuelle.
3/ On modifie la solution comme suit :
Répéter
Di := vrai :
Tantque (tour = j) ou (Dj=vrai) **Faire**
Tantque Dj = vrai **Faire** <rien>**Fait** ; tour :=i
Fait ;
<Section Critique i>
Di : = faux ;
Jusqu'à faux
a- Donner la condition d'entrée en SC.
b- Vérifier les 4 conditions de la SC.

Exercice 4 :
On suppose un système constitué uniquement de deux événements mémorisés e1, e2.
- Ecrire une implémentation des *deux primitives* suivantes :
- **Attendre (e1 et e2)** qui permet d'attendre que les événements e1 et e2 se produisent.
- **Déclencher (ei)** qui permet de réveiller tous les processus en attente de cet événement si leurs conditions sont satisfaites, dans ce cas, l'événement est acquitté. Si aucun processus ne l'attend, l'événement est mémorisé.

Exercice 5: On s'intéresse aux événements mémorisés. On définit les primitives suivantes :
Wait(k,e1,…en) : permet d'attendre *k* événements sur *n* déclenchés par des processus quelconques.
Signal(e) : déclenche l'arrivée de l'événement *e*. Tous les processus en attente de cet événement sont activés si leurs conditions sont satisfaites et l'événement est acquitté. Si aucun processus ne l'attend ou les conditions des processus en attente de l'événement ne sont pas satisfaites, l'événement est mémorisé.
Utiliser cet outil pour programmer l'application suivante dans deux cas différents :
1- Soient 3 processus ayant un point de rendez-vous. Un processus arrivant au point de rendez-vous continue son exécution :
a/ si les 2 autres sont arrivés à ce point.
b/ si au moins 1 autre est arrivé.
2- Proposer une implémentation de ces primitives

Exercice n°6 : Soit le graphe suivant :



1/ Ce graphe est-il proprement lié ?
2/ Exprimer ce graphe à l'aide de Parbegin/Parend et éventuellement avec les sémaphores.