

Exercice n°1 : (7 pts= 3 + 2 + 2)

Répondre aux questions suivantes :

- A- Quelle est la différence entre un moniteur classique et un moniteur avec conditions de Kessels ?
- B- Donner la différence entre une boîte aux lettres privée et une boîte aux lettres commune.
- C- Pourquoi les moniteurs sont qualifiés d'outils de synchronisation de haut niveau ?

Exercice 2 (5 pts)

- Donner le graphe de précédences associé au programme ci-dessous.

Programme P ;

Début

Se1, S13, S65 : Semaphore:=0;

Debut

Parbegin

Debut T6 ; V(S61) ; V(S65) Fin ;

Debut P(S13) ; T3 Fin;

Debut T4;

Parbegin Debut P(S61) ; T1 ; V(S13) Fin ; Debut P(S65) ; T5 Fin Parend ;

T2

Fin

Parend

Fin.

Exercice 3 : (8 pts= 5+ 3)

On s'intéresse à la gestion d'une classe de ressources (imprimante par exemple) à k exemplaires partagées par N processus. Chaque processus est connu par son identité i comprise entre 1 et N et peut demander un seul exemplaire de ressources à la fois.

Pour les processus en attente, la priorité est dans l'ordre inverse à l'ordre de blocage.

A) Réaliser cette gestion à l'aide des moniteurs classiques.

B) Donner une solution qui permet de satisfaire les processus dans l'ordre d'arrivée en utilisant les sémaphores.

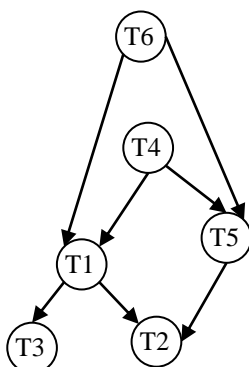
Bon Courage

Correction de l'Epreuve Finale

Exercice n°1 : (7 pts= 3 + 2 + 2)

- A- La différence entre un moniteur classique et un moniteur avec conditions de Kessels réside essentiellement au niveau des points de synchronisation.
- Dans la définition classique, la synchronisation se situe au niveau de deux points, l'un à l'exécution de *signal ()* et l'autre à l'exécution de *wait ()*. La condition de franchissement est décrite de manière symbolique. A l'exécution de *wait()*, le processus concerné se bloque de manière inconditionnelle. L'exécution de *signal* est nécessaire pour réveiller un processus bloqué sur une condition.
 - Dans un moniteur avec conditions de Kessels, la condition de franchissement est décrite de manière explicite en associant à *wait()* une condition booléenne *C*. Un processus se bloque sur *wait(C)* si la condition *C* n'est pas vérifiée. La primitive *Signal ()* n'est pas nécessaire étant donné que la réévaluation est faite de manière automatique, pour tous les processus bloqués, dès la sortie d'un processus du moniteur ou à son blocage par *wait ()*.
- B- La différence entre une boîte aux lettres privée et une boîte aux lettres commune réside au niveau de la propriété et aussi au niveau de la manipulation.
- Une boîte aux lettres privée est propriété d'un (groupe de) processus s'il est ou s'ils sont le/les seul(s) à l'utiliser en réception et qu'en réception. Elle est créée par une déclaration et disparaît à la fin du/des propriétaires.
 - Une boîte aux lettres commune est propriété du système d'exploitation, donc chaque processus peut l'utiliser en émission et en réception. Des primitives de création et de destruction sont alors nécessaires. Puisque cette BAL n'est propriété d'aucun processus, quand tous les processus qui lui font référence se terminent, celle-ci restera au niveau du système.
- C- Les moniteurs sont qualifiés d'outils de synchronisation de haut niveau car son expression est proche du langage de l'utilisateur et de plus il permet de décharger l'utilisateur de certains types de synchronisations, notamment la protection des variables de synchronisation déclarées au sein du moniteur.

Exercice 2 : (5 pts)



Exercice 3 :

A/ Solution avec moniteur classique :

M1 : Moniteur ;

Const N=..... ; k=.... ;

Var dispo, j : entier ; T : Tableau [1..N] de entier ;

c : Tableau [1..N] de condition;

entry procédure dem (i : entier) ;

Debut

Si (dispo=0) Alors j :=j+1 ; T[j] :=i ; c[i].wait () ; j :=j-1 Fsi ;

dispo :=dispo-1 ;

Fin ;

entry procédure lib () ;

Début

dispo :=dispo+1 ;

Si (j<>0) Alors c[T[j]].signal ()Fsi ;

Fin ;

Debut dispo :=k ; j :=0 Fin.

Remarque : On peut utiliser une File Explicite gérée de manière LIFO (ou une pile) au lieu du tableau T.

B/ Solution Fifo avec sémaphores :

Const N=.... ; k=... ;

Semaphore S :=k ;

Processus P (i : 1..N) ;

Debut

.....

P(S) ;

<Utiliser ressource>

V(S) ;

.....

Fin.