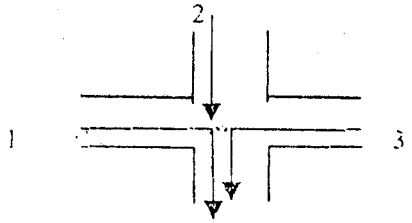
USTHB
Faculté d'Electronique & d'Informatique
Département Informatique

Année 01/02
Le 03/04/2002

EMD2(SYS02- SECTION A) Durée : 1 h 30 mn

Exercice 1 : (7 pts)

Etant donné un carrefour dont les sens possibles de circulation sont 1, 2, 3 (voir figure). La circulation se fait de manière continue et les *voitures* arrivent avec un nombre indéterminé dans chaque sens.
Un *agent* se place au milieu du carrefour et règle la circulation. Les priorités appliquées sont dans l'ordre suivant: priorité (1) > priorité (2) > priorité(3).



1/ Donner les différentes situations dans lesquelles l'agent peut être confronté et les actions qu'il doit envisager pour cela.
2/ Etablir la solution à ce problème à l'aide des sémaphores.

Exercice n°3 : (7 pts)

On suppose un système qui contient un nombre fixe n de processus et une seule classe de ressources à m instances. Chaque processus peut demander K exemplaires et la priorité est donnée à celui qui demande le moins d'instances. On désire gérer l'allocation de cette classe de ressources à l'aide des moniteurs classiques
1/ Donner les structures de données nécessaires et la forme générale des processus.
2/ Ecrire la solution correspondante.

Exercice 3: (6 pts)

Dans un système d'exploitation, on dispose des primitives de communication par boites aux lettres utilisant les primitives suivantes :

- Send (B, message) : Dépôt de message dans la boite aux lettres B.
- Receive (B, message) : Attente et retrait de message de la boite aux lettres B.

- Soit Le processus suivant:

```
Processus P ;
Struct m, mess : ..... ;
Début
    -
    Send( B1, ''vide'') ; Send (B2,''vide'') ;
    Répéter
        Receive(B1, m) ;
        Si (m<>''vide'')  Alors mess ← m
                          Sinon  Send (B1,''vide'') ;
                                 Receive (B2,m) ;
                                 Si (m<>''vide'') Alors mess ←m
                                                  Sinon  Send (B2,''vide'')
                                 Fsi ;
        Fsi ;
    Jusqu'à (mess<>''vide'') ;
    -
Fin.
```

B1 et B2 étant deux boites aux lettres communes à P et à d'autres processus.

1/ Que fait le processus P ?
2/ Expliquer le fonctionnement de la solution ?
3/ Si on replace la primitive Receive par la fonction read (B) qui retourne le premier message de la boite au lettre B s'il existe sinon vide, réécrire le processus P.

CORRECTION

(1)

Exercice n°1 :

1/

- Les différentes situations :

a) Pas des voitures
b) Au moins une voiture dans le sens i (i=1..3)
c) Au moins une voiture dans chaque sens 1 et 2
d) Au moins une voiture dans chaque sens 1 et 3
e) Au moins une voiture dans chaque sens 2 et 3
f) Au moins une voiture dans chaque sens 1 et 2 et 3

- Les actions à envisager :

a) rien faire.
b) faire passer les voitures du sens i une à une
c) faire passer les voitures du sens 1 une à une
d) faire passer les voitures du sens 1 une à une
e) faire passer les voitures du sens 2 une à une
f) faire passer les voitures du sens 1 une à une

2/ solution :

```
mutex : Tableau[1..3]de sémaphores :=1 ;
S     : Tableau[1..3]de sémaphores :=0 ;
Ag : sémaphore :=0 ;
nb : Tableau[1..3]de entier :=0 ; j : entier :=0 ;
```

```
Processus Voiture_sens(i : entier);
Début
    P(mutex[i]) ;
    nb[i] :=nb[i]+1 ;
    V(mutex[i])
    P(S[i]) ;
    <Traverser i >
    V(Ag) ;
Fin.
Fsi.
```

```
Processus Agent ;
Début
   Déb :j :=1
   Suite :P(mutex[j]) ;
         Si nb[j]<>0     Alors
           nb[j] :=nb[j]-1 ; V(mutex[j]) ;V(S[j]) ;
           Aller à Déb ;
                    Sinon
           V(mutex[j]) ; j :=(j+1); Si (j>3) Alors Aller à Déb
         Fsi ;
         Aller à Suite ;
Fin.
```

Mutex [i] : sémaphore d'exclusion mutuelle protège la variable partagée nb[i] entre l'agent et la voiture du sens i.
S[i] : sémaphore privé permet de bloquer chaque voiture du sens i en attendant le feu vert de l'agent
Ag : sémaphore privé permet de bloquer momentanément l'agent jusqu'à ce que la voiture en cours traverse le carrefour.
Nb[i] : compte le nombre de voitures du sens i en attente de la traversée.

Exercice n°2 :

1/ Les structures nécessaires et la forme générale d'un processus:

- F : liste dont chaque élément est un enregistrement contenant l'identité du processus ayant une demande pendante et le nombre d'exemplaires demandées. Cette liste est triée par ordre croissant du nombre d'instances demandées. A cette liste on associe les primitives suivantes :
  - Insérer(F, élément) insère élément dans la liste en conservant l'ordre croissant.
  - Extraire(F) : supprime le premier élément de la liste.
  - Vide(F) : retourne vrai si la liste est vide.
  - Premier(F) : retourne le premier élément de la liste F.

- S : tableau à n sémaphores privés dont S[i] est associé au processus Pi.
- Mutex : sémaphore d'exclusion mutuelle permettant de protéger les variables partagées
- Dispo : entier compte le nombre de ressources disponibles.

```
Processus P(i :entier) ;                    Le processus peut libérer par parties les ressources
Début                                       préalablement acquises.

        Demander(k)
        < Utiliser les ressources>
        Libérer(K)

Fin.
```

2/ La solution :

```
< Déclaration de la liste F et de ses procédures d'accès>
S : Tableau[1..n] de sémaphores :=0 ;
sémaphore Mutex :=1 ; entier Dispo :=m ;
```

Il s'agit d'écrire les primitives Demande (k : entier) et Libérer(k :entier) ;

```
Procédure Demander( k : entier) ;
Elément : enregistrement id, nb : entier fin ;
Début
        P(mutex) ;
                Si (Dispo<k)    Alors   élément.id :=i ; élément.k :=k ; insérer(F, élément) ;
                                        V(mutex) ; P(S[i])
                                Sinon  Dispo :=Dispo-k ;  V(mutex) ;
                Fsi ;
Fin ;

Procédure Libérer ( k : entier) ;
Début
        P(mutex) ;
                Dispo :=Dispo+k ;
                Tantque Non vide(F) et  (Dispo<=premier(F).nb)   Faire
                        Dispo :=Dispo - premier(F).nb ;
                        V(S[premier(F).id]) ; extraire(F) ;
                Fait ;
        V(mutex) ;
Fin ;
```

Ou

1/ Les structure nécessaires et la forme générale d'un processus:

- Nb : tableau à m entier Nb[i] indique le nombre de processus en attente de i ressources.
- S : tableau à m sémaphores privés dont S[i] bloque les processus demandant i ressources
- Mutex : sémaphore d'exclusion mutuelle permettant de protéger les variables partagées
- Dispo : entier compte le nombre de ressources disponibles.

```
Processus P(i :entier) ;                    Le processus peut libérer par parties les ressources
Début                                       préalablement acquises.

        Demander(k)
        < Utiliser les ressources>
        Libérer(K)

Fin
```