Exercice n°1 : (10 pts= 3 + 3 + 2 + 2)

Répondre aux questions suivantes :

A- Si on désire utiliser les moniteurs avec priorité pour l'allocation d'une ressource à N exemplaires identiques de telle sorte qu'on privilégie les processus qui demandent le plus d'instances, un processus non satisfait se bloque avec quelle valeur sur une condition donnée ? expliquer.

B- Comment résoudre le problème d'attente d'un seul message qui peut être envoyé exclusivement par un des deux processus P1 et P2 à l'aide du langage CSP?

C- Expliquer la notion de communication avec rendez-vous.

D- Quelle est la difficulté rencontrée quant à l'utilisation des événements dans la synchronisation de processus ?

Exercice 2 : (10 pts = 3 + 7)

On désire implémenter le modèle des Lecteurs/rédacteurs avec priorité aux rédacteurs en utilisant les moniteurs classiques.

A/ Lister les variables de synchronisation à utiliser en indiquant leurs rôles.

B/ Donner une solution à ce problème.

Bon Courage

Correction de l'Epreuve Finale

Exercice n°1 : (10 pts= 3 +3 + 2 + 2)

Répondre aux questions suivantes :

A- On les bloque sur c.wait (N-k) où k est le nombre de ressources demandées.
Le plus prioritaire est celui qui demande le plus de ressources et donc celui qui a la valeur (N-k) la plus petite qui correspond au processus réveillé par c.signal().

*B- *[P1 ?m1 → .... ; □ P2 ? m1→....]*

C- Le lien ne peut conserver temporairement aucun message donc l'émetteur et le récepteur doivent être prêt à communiquer simultanément (ie. chacun exécute respectivement sa primitive envoyer () ou recevoir ()) pour que la communication ait lieu.

D- Ils ne peuvent pas être généralisés à un problème général de synchronisation car ils sont rigides à manipuler.

Exercice 2 : (10 pts = 3 + 7)

Les variables et sémaphores

```
L: condition ;        // sert à l'attente des lecteurs.
E: condition;         // sert à l'attente des rédacteurs.
nla: entier;          // nombre de lecteurs en attente.
nlc: entier;          // nombre de lecteurs en cours.
nr:entier;            // nombre de lecteurs dans le système.
```

La solution

```
Lec_Red : Moniteur ;
Const N=.... ;
Var L, E : condition ;
nla, nlc, nr : entier ;
```

```
entree procedure dem_lec();
Debut
Si (nr<>0) Alors nla :=nla+1 ;
                 L.wait() ;
                nla :=nla-1
Fsi ;
nlc :=nlc+1
Fin ;
entree procedure fin_lec()
Debut
  nlc:=nlc-1;
  Si (ncl=0) Alors E.signal () Fsi
Fin.
```

```
entree procedure dem_ecr ();
Debut
nr :=nr+1
Si  (nlc<>0) ou (nr>1)
                Alors E.wait()
Fsi
Fin ;
```

```
entree procedure fin_ecr ();
Debut
nr :=nr-1 ;
Si (nr<>0)
      Alors E.signal ()
      Sinon
         Tant que (nla<>0) Faire
            nla:= nla-1;
            L.signal ()
         Fait
Fsi
Fin.
```

```
Initialisation
Debut
nla:=0 ; nlc:=0 ; nr:=0
Fin.
```