

## TD n°5 : Communication et synchronisation

### CORRECTION

#### Exercice 1 – Notions de cours

##### *Solution :*

- 1) Un autre processus peut accéder aux ressources partagées avant qu'un processus n'ait fini de les utiliser (modifier).
- 2) Oui, par exemple les sémaphores, les tubes et les files d'attente de messages.
- 3) Une suite d'instructions qui accèdent à des objets partagés avec d'autres processus.
- 4) B.

#### Exercice 2 – Sémaphores de synchronisation

##### *Solution :*

```
Semaphore 1vers2 = 0;
Semaphore 1vers3 = 0;
Semaphore 1vers4 = 0;
Semaphore 2vers5 = 0;
Semaphore 3vers5 = 0;
Semaphore 4vers6 = 0;
Semaphore 5vers6 = 0;
```

```
Process P1 {
    S1;
    1vers2.V();
    1vers3.V();
    1vers4.V();
}
```

```
Process P2 {
    1vers2.P();
    S2;
    2vers5.V();
}
```

```
Process P3 {
    1vers3.P();
    S3;
    3vers5.V();
}
```

```
Process P4 {
    1vers4.P();
    S4;
    4vers6.V();
}
```

```
Process P5 {
    2vers5.P();
    3vers5.P();
    S5;
    5vers6.V();
}
```

```
Process P6 {
    4vers6.P();
    5vers6.P();
    S6;
}
```

**Remarque :** seuls 3 sémaphores auraient pu suffir

### Exercice 3 – Sémaphores d'exclusion mutuelle

#### *Solution :*

1) Non, car si P2 est en section critique et P1 a exécuté P(mutex1) alors P1 est bloqué et empêche P3 d'entrer en section critique.

2) *Code du processus P1 :*

```
mutex1.P() ;
    n=n-1 ;
mutex1.V() ;
mutex2.P() ;
    Out = out +1 ;
mutex2.V() ;
```

### Exercice 4 – Lecteurs/rédacteurs

#### *Solution :*

1) Modèle des lecteurs et des rédacteurs (la voie joue le rôle de la base de données).

2)

```
autorisation = 1 ; //Protection de l'accès bidirectionnel à la voie
Sémaphore mutexAB = 1; //Protection du nb de trains utilisant la voie de A vers B
Sémaphore mutexBA = 1; //Protection du nb de trains utilisant la voie de B vers A
```

*Train AversB :*

```
mutexAB.P() //Nouveau train AversB
    si NbAB = 0
        alors autorisation.P()
    finsi
    NbAB=NbAB+1
mutexAB.V()
//Circulation sur la voie de A vers B
...
mutexAB.P() // Sortie de la voie par B
    Si NbAB = 1
        alors autorisation.V()
    finsi
    NbAB=NbAB-1
mutexAB.V()
```

*Train BversA :*

```
mutexBA.P() //Nouveau train BversA
    si NbBA = 0
        alors autorisation.P()
    finsi
    NbBA=NbBA+1
mutexBA.V()
//Circulation sur la voie de N vers A
...
mutexBA.P() // Sortie de la voie par A
    si NbBA = 1
        alors autorisation.V()
    finsi
    NbBA=NbBA-1
mutexBA.V()
```

## Exercice 5 – Producteurs/consommateurs

### *Solution :*

1)

```
Semaphore Mutex = 1, Plein = 0 ;
Message tampon[];
Producteur ( )
{
    Message m ;
    Tantque Vrai faire
        m = creermessage() ;
        Mutex.P() ;
        EcritureTampon(m) ;
        Mutex.V() ;
        Plein.V() ;
    FinTantque
}
Consommateur ( )
{
    Message m ;
    Tantque Vrai faire
        Plein.P() ;
        Mutex.P() ;
        m = LectureTampon() ;
        Mutex.V() ;
    Fin Tantque
}
```

2)

```
Semaphore Mutex = 1, Plein = 0, Vide = Max ;
Message tampon[Max];
Producteur ( )
{
    Message m ;
    Tantque Vrai faire
        Vide.P()
        m = creermessage() ;
        Mutex.P() ;
        EcritureTampon(m) ;
        Mutex.V() ;
        Plein.V() ;
    FinTantque
}
Consommateur ( )
{
    Message m ;
    Tantque Vrai faire
        Plein.P() ;
        Mutex.P() ;
        m = LectureTampon() ;
        Mutex.V() ;
        Vide.V() ;
    Fin Tantque
}
```