

Examen semestriel

Module de Systèmes d'exploitation

Corrigé

Exercice 1 : On reprend le problème du point de rendez-vous vu en TD. Proposez une solution à ce problème en utilisant les moniteurs de Hoare.

Réponse :

Structure du moniteur :

```
Type Rendez_Vous = Monitor
```

```
Const N = 10
```

```
Var
```

```
    I : integer ;
```

```
    X : Condition
```

```
Procedure Entry Arrivee
```

```
Begin
```

```
    I:= I + 1;
```

```
    If I = N Then X.Signal
```

```
End
```

```
Procedure Entry Franchir
```

```
Begin
```

```
    X.Wait;
```

```
End
```

```
Begin /* Initialisation */
```

```
    I := 0
```

```
End.
```

Structure des processus : RV étant une instance du type de moniteur Rendez_Vous.

Processus Pi	Processus Maitre
Début	Début
...	...
RV_Arrivee	RV_Franchir
...	...
Fin	Fin

(10 points)

Exercice 2 : On veut implémenter les sémaphores en utilisant les moniteurs de Hoare. Expliquez comment peut-on le faire. Donnez la structure du moniteur permettant d'implémenter les opérations wait(S) et signal(S) d'un sémaphore S, Précisez clairement les déclarations.

Structure du moniteur :

La structure du moniteur garantit l'accès en exclusion mutuelle des opérations *wait* et *signal*. Le blocage et le réveil d'un processus se fait par l'intermédiaire d'une variable conditionnelle (X dans notre corrigé). A partir de ces considérations, l'implémentation des sémaphores est une simple transcription de leur définition.

L'initialisation du sémaphore dépend du cas étudié (sémaphore d'exclusion mutuelle, de compatge, ... etc) ; la valeur initiale est un entier positif ou nul.

```
Type Semaphore = Monitor

Var   S : Integer ;
      X : Condition

Procedure Entry Wait( var S : Integer)
Begin
    S:= S - 1;
    If S < 0 Then X.Wait

End

Procedure Entry Signal( var S : Integer)
Begin
    S:= S + 1;
    If S <= 0 then X.Signal;
End

Begin /* Initialisation */
    S := N (N est un entier supérieur ou égal à 0)
End.
```

Soit SM une instance du moniteur Semaphore, la structure d'un processus utilisant les sémaphores est :

```
Processus P
Début
    ...
    SM.Wait(S)
    ...
    SM.Signal(S)
Fin
```

(10 points)