

TD N°2 Système d'exploitation  
Exclusion Mutuelle

**Exercice 1 :**

On se propose de programmer l'exclusion mutuelle entre deux tâches parallèles pour l'accès à une section critique : les seules opérations indivisibles sont l'affectation d'une valeur à une variable et le test de la valeur d'une variable.

Principe de la solution :

- Définir un ensemble de variables d'état, communes aux contextes des deux tâches.
- L'autorisation d'entrée en Section Critique sera définie par des tests sur ces variables, et l'attente éventuelle sera programmée comme une attente active (répétition cyclique des tests).

On supposera par la suite que les tâches sont cycliques et que leur comportement est le suivant :

```
While (true) {  
  Entrée_en_section_critique  
  Section_critique  
  Sortie_de_section_critique  
  Section_non_critique  
}
```

**Question 1**

On utilise une seule variable booléenne M telle que M = vrai si une des tâches se trouve dans sa section critique, faux sinon.

*Ecrire le programme*

*Vérifier que l'exclusion mutuelle ne peut être ainsi programmée.*

**Question 2**

On utilise une variable commune unique T telle que  $T = i$  si et seulement si la tâche  $P_i$  est autorisée à entrer en sa section critique ( $i = 0, 1$ ).

*Écrire le programme d'une tâche.*

*Montrer que la solution ne vérifie pas la condition c) (le blocage d'une tâche hors de sa section critique peut empêcher l'autre d'entrer en sa section critique) mais vérifie les autres conditions.*

### Question 3

On utilise  $C(i)$  variable booléenne attachée à la tâche  $P_i$  ( $i = 0, 1$ )

$C(i)$  = vrai si  $P_i$  est dans sa section critique ou demande à y entrer

$C(i)$  = faux si  $P_i$  est hors de sa section critique

$P_i$  peut lire et modifier  $C(i)$ , peut lire seulement  $C(j)$  si  $j$  différent de  $i$ .

Écrire le programme de la tâche  $P_i$

Vérifier qu'on ne peut pas obtenir la condition d'interblocage

### Question 4

On peut obtenir une solution correcte en combinant les solutions précédentes et en introduisant une variable supplémentaire  $T$  servant à régler les conflits à l'entrée de la section critique,  $T$  n'est modifiée qu'en fin de section critique.

L'ensemble des variables est:

- $C(i)$  avec la signification précédente (Question 3)
- $T$  avec la signification précédente (Question 2)

S'il y a conflit ( $C(i) = C(j) = \text{vrai}$ ),  $P_i$  et  $P_j$  exécutent une séquence d'attente où  $T$  a une valeur constante.

Si  $T = j$ , alors  $P_i$  annule sa demande en positionnant  $C(i)$  à faux,  $P_j$  peut alors entrer en section critique.  $P_i$  attend que  $T = i$  et refait sa demande en positionnant  $C(i)$  à vrai.

*Écrire le programme de  $P_i$ . Vérifier les 4 conditions*

**Exercice 2 :**

Vérifier les quatre conditions de l'exclusion mutuelle :

```

While{true}{
    Di = vrai ;

    While {Dj}{
        Si (tour ==j) alors
            Début
                Di=Faux ;

                While (tour ==j) ;

                Di=vrai ;
            FinSi
        }
    }

    <Section critique>

    Tour=j ;

    Di=Faux ;

}
    
```

**Exercice 3 :**

Vérifier les quatre conditions de l'exclusion mutuelle :

D1, D2 : booléen ; D1=D2=Faux ;  ParBegin P1 ; P2 ; ParEnd	Processus P1 : While{True}{ D1=vrai ; While{D2}{ D1=Faux ; D1=vrai ; <Section critique> D1=Faux ; } } )	Processus P2 : While{True}{ D2=vrai ; While{D1} D2=Faux ; D2=vrai ; <Section critique> D2=Faux ; } } )
------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------