

Exo_1 :

Soit les deux pseudo-codes de deux programme à exécuter en parallèle.

Soit « File » un fichier texte partagé.

```

Programme P1()
Début
  Write (FILE," Bienvenu ");
  WriteLn (FILE, " Mohamed ");
Fin ;
    
```

```

Programme P1()
Début
  Write (FILE, " Aurevoir " );
  WriteLn ( FILE, " Amir");
Fin ;
    
```

Question : Quels sont les différents possibilités d'écriture après une exécution en pseudo-parallèle (en concurrence) de P1 et P2.

Exo_2 :

Soit Nb_place variable partagé entre deux programmes P1 et P2 ;

Nb_place : entier ; Nb_place ← 1 ;

```

Programme P1()
Début
  Si Nb_place > 0 alors
    Nb_place ← Nb_place -1;
    <Réserver une (01) place >
Fin ;
    
```

```

Programme P2()
Début
  Si Nb_place > 0 alors
    Nb_place ← Nb_place -1;
    < Réserver une (01) place >
Fin ;
    
```

On considère une exécution en concurrence (parallèle ou en pseudo-parallèle) de P1 et P2

1. Quel la valeur finale de Nb_place ?
2. Quel est état logique du résultat (bon ou faux) ?

Exo_3:

En utilisant les sémaphores, modifier les programmes de trois processus concurrents P1, P2 et P3 pour contrôler les accès aux variables partagées : "in" et "out".

<pre> Processus P1 { ... out=out+1 ; in=in-1 ; ... } </pre>	<pre> Processus P2 { ... out=out-1 ; ... } </pre>	<pre> Processus P3 { ... in=in+1 ; ... } </pre>
---	---	---

Exo_4:

Soient trois processus concurrents P1, P2 et P3 qui partagent trois variables initialisé à zéro : a, b et c. Pour contrôler les accès à ces variables, un programmeur propose les codes ci-dessous : Sémaphore MUTEX1=1, MUTEX2=1, MUTEX3=1;

PROCESSUS P1	PROCESSUS P2	PROCESSUS P3
P(MUTEX1) ; a←a+1; P(MUTEX2) ; b=b+1 ; V(MUTEX2) ; V(Mutex1) ;	P(MUTEX2) ; b←b-1; P(MUTEX3) ; c←c-1 ; V(MUTEX3) ; V(MUTEX2) ;	P(mutex3) ; c←c+1; P(MUTEX1) ; a←a-1 ; V(MUTEX1) ; V(mutex3) ;

Question : Cette proposition est-elle correcte (justifier votre réponse) ?
Sinon Proposer une solution correcte respectant les conditions de Dijkstra?

Exo 5:

Certaines sous-expressions d'une expression arithmétique peuvent être évaluées dans un ordre quelconque. On peut donc les évaluer en parallèle, si on dispose d'un nombre suffisant de processeurs. Soit l'expression : $(a + b) * (c + d) - (e/f) . 1$.

Question : donner les programmes de des trois processus p1, p2, p3, tel que P2 calcule $c + d$, P3 calcule e/f et P1 le résultat. Attention, P1 doit attendre la fin de P2 et P3.

Exo 6:

Soient 2 processus séquentiels qui exécutent les programmes suivants :

Processus P1

Debut

Répéter Tant que « Vrai »

I1;

Fin Tant que

Fin

Processus P2

Debut

Répéter Tant que « Vrai »

I2;

Fin Tant que

Fin

Question : Utilisez deux sémaphores pour synchroniser les 2 processus de telle manière que les instructions I1 (de P1) et I2 (de P2) se déroulent toujours dans l'ordre : **I2 I1 I2 I1 I2 I1 I2 I1 I2 I1 I2**

Exo 7:

Pour fabriquer un objet X on a besoin de 2 pièces A et B. on dispose de trois processus P1, P2 et P3.

- P1 fabrique une pièce **A** et la dépose dans le bac B1.
- P2 fabrique une pièce **B** et la dipose dans le bac B2.

- P3 retire une pièce A de Bac B1 et une pièce B de B2 et assemble ces deux pièce

Question : En utilisant les sémaphores écrire les programmes des trois processus.

Exo 8:

Deux villes A et B sont reliés par une seule voie de chemin de fer. Les trains peuvent circuler dans le même sens de A vers B ou de B vers A. Mais, ils ne peuvent pas circuler dans les sens opposés. On considère deux classes de processus : les trains allant de A vers B (Train AversB) et les trains allant de B vers A (Train BversA). Ces processus se décrivent comme suit :

Train AversB :

Demande d'accès à la voie par A ;
Circulation sur la voie de A vers B;
Sortie de la voie par B;

Train BversA :

Demande d'accès à la voie par B ;
Circulation sur la voie de B vers A;
Sortie de la voie par A;

Question : Écrire sous forme de commentaires en utilisant les sémaphores, les opérations P et V, les codes de demandes d'accès et de sorties, de façon à ce que les processus respectent les règles de circulation sur la voie unique.

Exo 9:

En utilisant des sémaphores, écrire les programmes de rendez-vous de N processus dans ces cas :

- cas 1: $N=2$:
- cas 2 : $N=3$;
- cas 3 : $N>3$;