Faculté des Sciences 3ème Année licence
Département d'informatique TD n°1, Matière SE2

Exercice 1 : Soit la variable x partagée entre les thread A et thread B dont les bouts d'instructions sont les suivants :

Thread A	Thread B
a1 x = 5	b1 x = 7
a2 print x	

Quelle est la valeur de x imprimée ? Quelle est la valeur finale de x quand toutes ces instructions sont exécutées ? ça dépend de l'ordre dans lequel les instructions sont exécutées, appelé le chemin d'exécution. Un chemin possible est a1< a2< b1, auquel cas la valeur de x affichée (imprimée) est 5, mais la valeur finale de x est 7.

1) Quel est le chemin d'exécution où la valeur affichée est 5 et la valeur finale est 5 ?
2) Quel est le chemin d'exécution où la valeur affichée est 7 et la valeur finale est 7 ?
3) Y-a-t-il un chemin menant à l'affichage de la valeur 7 et valeur finale de x 5? Justifier votre réponse.

Exercice 2 : On considère l'algorithme suivant d'exclusion mutuelle pour deux processus P1 et P2, où les variables locales *p1_dedans* et *p2_dedans* appartenant respectivement à P1 et P2 sont initialisés à faux.

```
P1 : debut                          P2 : debut
       …..                                 …..
   A :  Tant que p2_dedans faire       A :  Tant que p1_dedans faire
          Aller à A                           Aller à A
        Ftantque                            Ftantque
      p1_dedans := vrai;                  P2_dedans := vrai;
        <section critique 1 >               <section critique 2 >
      P1_dedans := faux ;                 P2_dedans := faux ;
      ….                                  ….
    Fin                                 Fin
```

1. Montrer que cette solution est incorrecte. Quel est le problème ?
2. Donner une solution qui permet de résoudre ce problème.
3. Donner une solution correcte (l'algorithme de Dekker)

Exercice 3 : Soit un système composé de 3 processus P1, P2, P3 et 2 tampons tamp1 et tamp2 contenant une seule case chacun. Le processus P1 exécute une requête et transmet (via tamp1) le résultat au processus P2. Le processus P2 prélève ce résultat et effectue un traitement puis dépose dans tamp2 le résultat obtenu. Le processus P3 prélève le contenu de tamp2 et l'imprime.
Question : Donner les algorithmes des 3 processus permettant leur synchronisation.

Exercice 4 : Le code ci-dessous consiste en deux processus qui utilisent des sémaphores binaires.
Identifiez une séquence (chemin) d'exécution et de changements de contexte qui amènent ces deux processus dans une situation d'interblocage (deadlock).

Var mutex, data: Semaphore binaire,

```
P1                              P2
Debut                           debut
  ….                              …..
 P(mutex);                       P(data);
/* faire des choses */           P(mutex);
 P(data);                        /* faire d'autres choses */
 /* faire quelque chose */       V(mutex);
 V(data);                        V(data); }
 V(mutex);                        ……
  …..                           Fin
Fin
```

Question : Expliquez ce qui amène ces deux processus dans un deadlock. Proposez des modifications du code qui rend impossible tout interblocage (deadlock). En tirer un principe général à respecter pour éviter les deadlocks.

Exercice 5: Soient trois processus P1, P2, P3 tels que : P1 calcule la somme S1= A+B et P2 calcule S2= A+4 puis P3 calcule S3=S1+S2 ; écrire les codes des trois processus en utilisant les sémaphores.