



Durée : 1h45 – Documents autorisés

Threads & Sémaphores – (9 points)

1- Les trois threads, A, B et C suivantes sont exécutées de manière concurrente ; elles utilisent pour leur synchronisation deux sémaphores S1 et S2.

La valeur initiale de S1 est 1, celle de S2 est 0 et celle de x est 0.

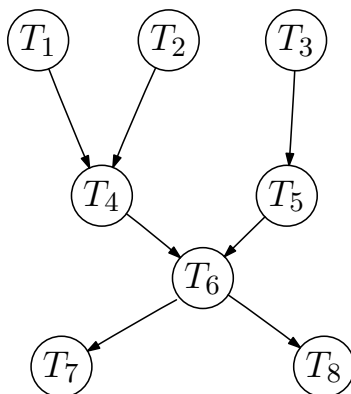
- a. Quelles sont les différentes valeurs possibles de la variable x à la fin de l'exécution complète des trois threads?
b. Qu'est-ce que cela changerait si la valeur initiale de S1 était 2 au lieu de 1?

Thread A { P(s2); x = x * 2; V(s1); }

Thread B { P(s1); x = x * x; V(s1); }

Thread C { P(s1); x = x + 3; V(s2); }

2- Écrire un algorithme pour résoudre le problème suivant (vous pouvez vous inspirer de la notation de l'exercice 1):



Un programme a été décomposé en 8 threads {T1, T2, T3, T4, T5, T6, T7, T8}. L'analyse du parallélisme pouvant être exploité entre ces différentes threads est la suivante:

- les threads {T1, T2, T3} peuvent être réalisées en parallèle;
- la thread T4 ne peut être faite qu'après les threads {T1, T2};
- la thread T5 ne peut être faite qu'après la thread {T3};
- les threads T4 et T5 peuvent s'exécuter en parallèle;
- la thread T6 ne peut être faite qu'après les threads {T4, T5};
- les threads T7 et T8 ne peuvent être exécutées qu'après la thread T6;
- les threads T7 et T8 peuvent s'exécuter en parallèle.

Question:

- a. Combien de Sémaphores faudra-t-il pour réaliser ce travail?
b. Donnez l'algorithme qui permet de synchroniser les threads entre elles.

■ ■ ■ ■ Création de processus avec fork – (3 points)

3– Indiquez ce qui va être affiché à l'exécution du programme, dont le source est le suivant :

3pts

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void main(void)
5 {
6     int x = 1; int resultat;
7
8     resultat = fork();
9     if (resultat)
10    {
11        x = x * 2;
12        printf("Je suis le processus A et x vaut %d", x);
13    }
14    else
15    {
16        x = x + 3;
17        printf("Je suis le processus B et x vaut %d", x);
18        resultat = fork();
19        if (resultat)
20        {
21            x = x + 5;
22            printf("Je suis le processus C et x vaut %d", x);
23        }
24    }
25    x = 3 * x;
26    printf("Je suis le processus D et x vaut %d", x);
27 }
```

■ ■ ■ ■ Signaux & tubes vs threads – (8 points)

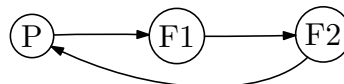
4– Écrire le programme réalisant le travail suivant :

4pts

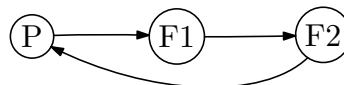
- soit le processus P:
 - ◊ il crée deux fils F1 & F2;
 - ◊ il communique à l'aide d'un tube avec F1;
- F1 communique à l'aide d'un tube avec F2;



- On effectue 10 fois le travail suivant :
 - ◊ P envoie à F1 la valeur 1 et attend;
 - ◊ F1 envoie à F2 la valeur 1 et attend;
 - ◊ F2 affiche la valeur 1;
 - ◊ F2 envoie un signal à P pour le faire recommencer :



- ◊ P envoie à F1 la valeur 2 et attend;
- ◊ F1 envoie à F2 la valeur 2 et attend;
- ◊ F2 affiche la valeur 2;
- ◊ F2 envoie un signal à P pour le faire recommencer :



- ◊ etc.

5– Donnez une solution à base de threads et de sémaphores à l'exercice 4 en vous inspirant des

4pts TDs/TP.