



Durée : 2h — Documents autorisés

■■■■ **Threads & Sémaphores – (12 points)**

10pts 1– Une société de production de série télévisée vous contacte afin de lui fournir un système réaliste pour la gestion d'une prison de haute sécurité futuriste qu'elle pourra ensuite utiliser dans l'écriture de ses scénarios, et même peut-être l'utiliser comme sujet principal pour une série.

Le scénario de base pour la prison est le suivant :

- ▷ chaque prisonnier est muni d'un bracelet qu'il ne peut enlever et qui va embarquer un ordinateur exécutant le programme que vous allez concevoir ;
- ▷ il existe deux classes de prisonniers A et B qui ont des autorisations d'accès différentes aux différentes parties de la prison :
 - ◇ la classe A de 5 prisonniers très astucieux et capables de s'évader qui doivent être gérés avec des restrictions ;
 - ◇ la classe B de 30 prisonniers moins astucieux ;
- ▷ la prison possède :
 - ◇ 15 cellules : 5 individuelles et 10 pouvant recevoir 3 prisonniers ;
 - ◇ un court de tennis pour un ou deux joueurs (quand un joueur est seul il s'entraîne contre un mur de rebond) ;
 - ◇ un réfectoire de 35 places ;
 - ◇ un lieu extérieur de promenade limité à 5 prisonniers au plus ;

Dans le fonctionnement normal de la prison, chaque prisonnier se voit attribuer une cellule unique identifiée par son numéro de 1 à 5 pour les individuelles et de 6 à 15 pour celles de 3. À la fin de la journée les prisonniers doivent aller obligatoirement dans leur cellule et ne pas se trouver dans les parties communes.

Chaque cellule n'est accessible qu'au prisonnier dont le bracelet l'autorise.

Questions :

- a. pour le premier scénario, les producteurs envisagent de montrer le fonctionnement de la prison (2pts) dans son mode normal :
 - ◇ chaque prisonnier de classe A accède uniquement à sa cellule individuelle ;
 - ◇ les prisonniers de classe B accède par groupe de 3 à la même cellule qu'ils partagent ;Décrivez votre solution en termes de Sémaphores pour gérer l'accès des prisonniers de classe A et B à leur cellules respectives.
- b. Pour un autre scénario, les producteurs veulent montrer le fonctionnement du court de tennis : (2pts)
 - ◇ deux prisonniers de classe B peuvent avoir accès ensemble au court ;
 - ◇ un seul prisonnier de classe A peut accéder au court de tennis ;Décrivez votre solution en termes de Sémaphores pour gérer l'accès des prisonniers de classe A et B au court de tennis.
- c. Le scénario suivant porte sur le lieu de promenade dont l'accès est réglementé de la façon suivante : (2pts)
 - ◇ seul un seul prisonnier de classe A peut s'y trouver simultanément avec d'autres prisonniers de classe B ;
 - ◇ les prisonniers de classe B peuvent y avoir accès comme ils veulent à condition qu'il n'y ait pas plus de 5 prisonniers ;Décrivez votre solution en termes de Sémaphores pour gérer l'accès des prisonniers de classe A et B à la promenade.

Est-ce que votre solution est « équitable » ?

d. Le fonctionnement du réfectoire est le suivant : tous les prisonniers peuvent y avoir accès librement. (2pts)
Décrivez votre solution en terme de Sémaphores pour gérer l'accès des prisonniers de classe A et B au réfectoire.

e. Pour un nouveau scenario, une restriction a été mise en place suite à une tentative d'évasion : (2pts)
◇ le réfectoire n'est accessible simultanément qu'aux prisonniers de la même classe : un prisonnier d'une classe peut entrer dans le réfectoire uniquement si le réfectoire est vide ou occupé par des prisonniers de même classe.

Décrivez votre solution en termes de Sémaphores pour gérer l'accès des prisonniers de classe A et B à la promenade.

Est-ce que votre solution est « équitable » ?

2– Questions sur l'utilisation des threads :

2pts a. Pourquoi y-a-t-il des **risques de corruption** de données ? (0,5pt)

b. Comment est partagée la « **pile ou stack** » entre les threads ? (0,5pt)

Questions sur l'utilisation des sémaphores :

d. Comment sont réglés les problèmes de **famine** et d'**équité** ? (1pt)

■■■■ Signaux & tubes & fork – (8 points)

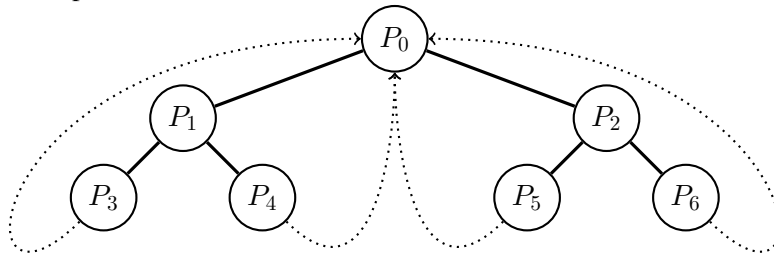
3– Soit le programme C suivant :

2pts

```
1 #include <stdio.h>
2
3 void traitement(int a, int b)
4 {
5     printf("-->%d\n", a*b);
6     if (fork())
7     {
8         if (a)
9         {
10            traitement(a-1, 2);
11        }
12    }
13 }
14 int main()
15 {
16     int a = 3;
17     int b = 1;
18
19     traitement(3,1);
20     printf("-->%d\n", a*b);
21 }
```

Que va-t-il afficher lors de son exécution ?

4– On veut mettre en place une structure **aborescente** de travail avec des tubes permettant de communi-
6pts quer entre processus d'un **niveau** de l'arbre vers le suivant :



- le trait en gras indique un « tube » de communication ;
- le trait pointillé indique l'envoi d'un « signal ».

1. un processus P_0 crée 2 fils $\{P_1, P_2\}$;
2. il crée un tube avec ses fils P_1 et P_2 permettant de communiquer du père vers le fils ;
3. le fils P_1 va :
 - ◇ créer 2 fils $\{P_3, P_4\}$;
 - ◇ établir un tube permettant de communiquer avec chacun de ses fils (de P_1 vers P_3 et de P_1 vers P_4) ;
 - ◇ P_3 et P_4 vont, après la mise en place du tube avec leur parent, envoyer un **signal à P_0** ;
4. le fils P_2 va :
 - ◇ créer 2 fils $\{P_5, P_6\}$;
 - ◇ établir un tube permettant de communiquer avec chacun de ses fils (de P_2 vers P_5 et de P_2 vers P_6) ;
 - ◇ P_5 et P_6 vont, après la mise en place du tube avec leur parent, envoyer un **signal à P_0** ;
5. lorsque le père reçoit les 4 signaux (un de chacun de ses « petits fils »), il exécutera la fonction `void travail_arbre()` qui sera fournie dans une bibliothèque externe à votre programme.

Questions :

- a. Pourquoi est-il nécessaire d'envoyer un **signal** depuis les « petits fils » vers le parent avant d'exécuter la fonction `travail_arbre` ? (1pt)
- b. Comment les processus enfants de P_1 et P_2 vont pouvoir connaître le *pid* de P_0 ? (1pt)
- c. Est-il possible d'avoir un *code* similaire pour le travail de P_1 et P_2 ? (1pt)
- d. Écrire le programme C réalisant **uniquement** la création des 7 processus P_i de manière à permettre le travail demandé : (3pts)
 - ◇ vous n'avez pas à créer de tubes, ni à envoyer de signal ;
 - ◇ vous devez gérer les Pids (mémoriser, partager) pour rendre l'intégration du code des signaux et tubes possible.