



Durée : 2h — Documents autorisés

■ ■ ■ Threads & Sémaphores – (7 points)

1– Une école d'ingénieurs dont la spécialité est la sécurité de l'information veut mettre en place un système automatique et sécurisé pour le passage des examens de TP :

- chaque étudiant est muni de son smartphone sur lequel peut être installé un logiciel développé par l'école ;
- chacun de ces smartphones possède un dispositif radio permettant d'interagir avec la porte de la salle où va se dérouler les examens et qui permet de laisser entrer ou non l'étudiant ;
- un serveur central gère :
 - ◇ les différents postes informatiques où l'étudiant peut consulter le sujet de l'épreuve ainsi que réaliser son travail ;
 - ◇ le déblocage de la porte en interaction avec le logiciel installé dans le smartphone.

Questions :

Dans la formation, il existe 4 groupes de 12 étudiants qui vont passer un examen dans une salle de TP équipée de 12 postes informatiques.

On voudrait automatiser la remise du sujet du projet sur lequel ils vont travailler :

- a. (1pt)
- ◇ Il existe un sujet pour chaque groupe ;
 - ◇ le serveur central ne doit installer le sujet qu'une fois que les 12 étudiants d'un même groupe sont entrés dans la salle ;
 - ◇ les 4 examens, un pour chaque groupe se succèdent dans la salle.

Proposez un programme à base de sémaphore mettant en œuvre votre solution en considérant une exécution concurrente entre le smartphone d'un étudiant et le serveur central.

- b. (1pt)
- On veut empêcher maintenant que les étudiants de deux groupes différents accèdent simultanément à la salle, c-à-d qu'un étudiant du groupe suivant puisse entrer alors qu'il reste un étudiant du groupe précédent.

Modifiez votre proposition de la question précédente pour en tenir compte.

- c. (2pts)
- Pour cette épreuve, les étudiants d'un même groupe travaillent en trinôme, et il existe un sujet par trinôme.

- ◇ Le sujet d'un trinôme ne peut être téléchargé que lorsque tous les étudiants du même trinôme sont entrés.
- ◇ On ne passe d'un groupe de 12 étudiants à un autre que lorsqu'ils ont tous fini de travailler en trinôme.
- ◇ Les différents trinômes peuvent venir travailler dans n'importe quel ordre.

Proposez une nouvelle solution tenant compte de ces contraintes.

- d. (2pts)
- L'École voudrait maintenant que la durée de l'épreuve en trinôme soit comptabilisée :

- ◇ l'étudiant d'un trinôme qui arrive le dernier déclenche l'obtention du sujet et la mémorisation de l'heure de début du travail ;
- ◇ l'étudiant d'un trinôme qui part le dernier déclenche la mémorisation de l'heure de fin du travail.

Proposez une nouvelle solution tenant compte de ces contraintes.

- e. (1pt)
- On voudrait que le dernier étudiant du dernier groupe quittant la salle déclenche l'extinction du serveur central.

Comment peut-on ajouter cette fonctionnalité ?

■ ■ ■ Programmation asynchrone — (9 points)

2– On voudrait disposer d'un mécanisme de Sémaphore dans un programme écrit pour Node.js.

9pts

- a. Résumez comment fonctionne une sémaphore dans un programme en langage C classique utilisant des threads POSIX :
 - I. Comment une Sémaphore est identifiée du point de vue du programme ? (2pts)
 - II. Est-ce qu'il existe de l'« attente active », lorsqu'une thread demande une sémaphore qui n'est plus disponible ?
 - III. Comment peut-on prendre ou libérer une sémaphore plusieurs fois ?
 - IV. Comment fonctionne l'ordonnancement des threads entre : threads suspendues en attente de sémaphore non encore disponible, threads non suspendues, threads suspendues en attente de sémaphore devenant disponible ?
- b. Donner une code pour Node.js permettant de : (3pts)
 - ◇ définir une sémaphore ;
 - ◇ libérer ou prendre une sémaphore ;
 - ◇ interagir avec l'ordonnancement ;
- c. Vous justifierez votre solution en écrivant et en commentant un programme qui : (2pts)
 - ◇ fonctionne dans le cas du problème du producteur/consommateur avec un buffer de 2 cases ;
 - ◇ garantit un fonctionnement aléatoire et équitable entre celui du producteur et du consommateur ;
- d. Avec votre solution est-ce qu'il y a des risques de corruption de données lors de l'utilisation de vos sémaphores ? (1pt)

■ ■ ■ Création de processus — (3 points)

3– Soit le programme C suivant :

3pts

```
1 | #include <stdio.h>
2 | void fonction(int val)
3 | { int i=0; int resultat = 0;
4 |
5 |     for(i=0; i < 2; i++)
6 |     { resultat = fork();
7 |       if (resultat)
8 |         { val = val * 3;
9 |           printf("Je suis le processus A et val=%d\n", val);
10 |        }
11 |     }
12 | }
13 |
14 | int main()
15 | { int resultat = 0;
16 |   resultat = fork();
17 |
18 |   if (resultat)
19 |   {
20 |     fonction(3);
21 |   }
22 |   fonction(2);
23 | }
```

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

■ ■ ■ Programmation concurrente — (2 points)

4– Est-ce que le problème d'équité est plus simple à résoudre en programmation asynchrone ou en programmation à base de threads Posix ?

2pts

Vous justifierez votre réponse.