



Université Toulouse III – Paul Sabatier  
118 route de Narbonne  
31062 Toulouse cedex 9

---

## Travaux pratiques – n°4

### Threads Posix et verrous d'exclusion mutuelle

#### Exercice noté

---

Le but de cet exercice est de gérer la réservation de N places de concert qu'une salle met à disposition de ses spectateurs. Une place ne peut être occupée que par un spectateur à la fois. Dans la question 2, on suppose aussi qu'il existe NCP catégories de places (parterre, vip, devant de scène, aile gauche, aile droite...) et que dans chaque catégorie, il y a le même nombre NP de places.

Un spectateur va demander à obtenir une réservation pour NPR ( $\leq$  NP) places avant de pouvoir en profiter et les libère lorsqu'il a fini de les utiliser.

On veut écrire l'équivalent des opérations proposées par un moniteur de Hoare pour synchroniser les accès à ces places de manière cohérente, et qui seraient spécifiées ainsi :

```
Moniteur GestionPlaces {  
    // Obtenir la réservation de nbP places de la catégorie cPlaces  
    reserverPlaces (int cPlaces, int nbP) ;  
    // Rendre les nbP places de la catégorie cPlaces utilisées  
    libererPlaces (int cPlaces, int nbP) ;  
}
```

On représente un spectateur par un thread Posix ayant le comportement suivant :

```
spectateur (monId) {  
    TantQue (je n'ai pas fait NResa réservations) {  
        . . . . .  
        // Je veux utiliser nbP places de la catégorie cP  
        reserverPlaces(cP, nbP) ;  
        // J'utilise les nbP places qui m'ont été allouées  
        . . . . .  
        // J'ai fini d'utiliser les nbP places, je les libère  
        libererPlaces(cP, nbP) ;  
    }  
}
```

À partir du squelette de code exoTp4.c fourni sur Moodle, programmer une application pour chacune des questions 1 et 2 ci-après.

#### Attention :

Vous devrez **rendre un programme pour les questions 1 et 2** dans les dépôts prévus à cet effet.

Vous n'apporterez des **modifications qu'aux endroits spécifiés** explicitement en commentaire dans le code fourni. Le **reste du code (y compris les prototypes des opérations demandées) doit rester inchangé** (au **risque de rendre vos réponses invalides**).

Des indications supplémentaires sont fournies après les questions, ainsi qu'un exemple d'exécution.

---

**Question 1**

---

Il n'y a qu'**une seule catégories de places** (cela revient à ne pas utiliser les paramètres correspondants du main fourni et des fonctions à écrire).

L'application doit permettre à un nombre NS de spectateurs de partager de manière cohérente l'utilisation de NP places (NS, NP ainsi que le nombre de réservations successives effectuées par un spectateur constituent les paramètres de l'application) en servant les demandes dans **l'ordre de leur arrivée** (voir Indications ci-dessous).

---

**Question 2**

---

On considère maintenant **NCP catégories de places** (il faut alors tenir compte du paramètre correspondant dans le main fourni).

On suppose qu'un spectateur spécifie, lors de ses demandes de réservation, la catégorie qu'il souhaite réserver. Les requêtes sont toujours servies dans **l'ordre de leur arrivée** (voir Indications ci-dessous).

Modifier votre application pour prendre en compte cette nouvelle contrainte.

---

**Question 3**

---

Dans le cas de la question 2, sans le programmer, et en quelques mots, que proposeriez-vous pour assurer un parallélisme maximal ?

Répondre en commentaires au début du code déposé pour la question 2.

---

**Indications**

---

Attention : N'oubliez pas qu'il existe des différences entre les conditions Posix et les « variables conditions » des Moniteurs de Hoare étudiées en Cours-TD. Outre les différentes déjà énoncées en TP, on rappelle que :

- Les conditions Posix ne supportent pas l'équivalent de l'opération wait() à priorité ou de l'opération empty() qui ont été utilisées en Cours-TD.
- L'opération pthread\_cond\_signal() réveille un des threads bloqués sur la variable condition, sans être certain de réveiller le premier qui a été bloqué.

Dans cet exercice, « l'ordre d'arrivée » signifie que la demande du premier spectateur réveillé est prioritaire sur celles des autres. S'il ne peut être satisfait (car demandant trop de places), sa demande est mémorisée et il est bloqué seul en attente de pouvoir être servi ; les suivants étant bloqués par ailleurs. Ce « premier » sera débloqué en priorité dès que le nombre de places demandé deviendra disponible.

Toutefois, à cause de la différence notée ci-dessus, il se peut qu'avec les conditions Posix, l'ordre ne soit pas totalement respecté (on ne vous demande pas de faire mieux pour cet exercice).

## Exemples d'exécution liés à la question 2

---

- Le principe général est le même pour la question 1 puisqu'une seule catégorie de places est considéré dans les exemples.
- Ici, 8 spectateurs demandent 1 fois un certain nombre de places parmi les 4 disponibles (dans la seule classe existante).

`./exoTP4 1 4 8 1`

```

Spectateur 2 (2058715470) : (resa 1) veut reserver 3 places de categorie 0
Spectateur 2 (2058715470) : (resa 1) obtient      3 places de categorie 0
Spectateur 0 (2075500856) : (resa 1) veut reserver 3 places de categorie 0
Spectateur 0 (2075500856) : (resa 1) je serai 1er pour reserver 3 places de categ. 0
Spectateur 2 (2058715470) : (resa 1) occupe      3 places de categorie 0
Spectateur 4 (2041930074) : (resa 1) veut reserver 3 places de categorie 0
Spectateur 4 (2041930074) : (resa 1) pas le 1er pour reserver 3 places de categorie 0
Spectateur 5 (2033537356) : (resa 1) veut reserver 4 places de categorie 0
Spectateur 5 (2033537356) : (resa 1) pas le 1er pour reserver 4 places de categorie 0
Spectateur 1 (2067108162) : (resa 1) veut reserver 4 places de categorie 0
Spectateur 1 (2067108162) : (resa 1) pas le 1er pour reserver 4 places de categorie 0
Spectateur 3 (2050322750) : (resa 1) veut reserver 4 places de categorie 0
Spectateur 3 (2050322750) : (resa 1) pas le 1er pour reserver 4 places de categorie 0
Spectateur 6 (2025144648) : (resa 1) veut reserver 3 places de categorie 0
Spectateur 6 (2025144648) : (resa 1) pas le 1er pour reserver 3 places de categorie 0
Spectateur 7 (2016751936) : (resa 1) veut reserver 4 places de categorie 0
Spectateur 7 (2016751936) : (resa 1) pas le 1er pour reserver 4 places de categorie 0
Spectateur 2 (2058715470) : (resa 1) a libere      3 places de categorie 0
Spectateur 0 (2075500856) : (resa 1) obtient      3 places de categorie 0
Spectateur 0 (2075500856) : (resa 1) occupe      3 places de categorie 0
Spectateur 4 (2041930074) : (resa 1) je passe 1er pour reserver 3 places de categ. 0
Spectateur 0 (2075500856) : (resa 1) a libere      3 places de categorie 0
Spectateur 4 (2041930074) : (resa 1) obtient      3 places de categorie 0
Spectateur 4 (2041930074) : (resa 1) occupe      3 places de categorie 0
Spectateur 5 (2033537356) : (resa 1) je passe 1er pour reserver 4 places de categ. 0
Spectateur 4 (2041930074) : (resa 1) a libere      3 places de categorie 0
Spectateur 5 (2033537356) : (resa 1) obtient      4 places de categorie 0
Spectateur 5 (2033537356) : (resa 1) occupe      4 places de categorie 0
Spectateur 1 (2067108162) : (resa 1) je passe 1er pour reserver 4 places de categ. 0
Spectateur 5 (2033537356) : (resa 1) a libere      4 places de categorie 0
Spectateur 1 (2067108162) : (resa 1) obtient      4 places de categorie 0
Spectateur 1 (2067108162) : (resa 1) occupe      4 places de categorie 0
Spectateur 3 (2050322750) : (resa 1) je passe 1er pour reserver 4 places de categ. 0
Spectateur 1 (2067108162) : (resa 1) a libere      4 places de categorie 0
Spectateur 3 (2050322750) : (resa 1) obtient      4 places de categorie 0
Spectateur 3 (2050322750) : (resa 1) occupe      4 places de categorie 0
Spectateur 6 (2025144648) : (resa 1) je passe 1er pour reserver 3 places de categ. 0
Spectateur 3 (2050322750) : (resa 1) a libere      4 places de categorie 0
Spectateur 6 (2025144648) : (resa 1) obtient      3 places de categorie 0
Spectateur 6 (2025144648) : (resa 1) occupe      3 places de categorie 0
Spectateur 7 (2016751936) : (resa 1) je passe 1er pour reserver 4 places de categ. 0
Spectateur 6 (2025144648) : (resa 1) a libere      3 places de categorie 0
Spectateur 7 (2016751936) : (resa 1) obtient      4 places de categorie 0
Spectateur 7 (2016751936) : (resa 1) occupe      4 places de categorie 0
Spectateur 7 (2016751936) : (resa 1) a libere      4 places de categorie 0
Fin de l'application de gestion des places

```

- Ici, 3 spectateurs demandent 2 fois un certain nombre de places d'une certaine catégorie parmi 2, comptant chacune 3 places.

./exoTP4 2 3 3 2

Spectateur 0 (3075832640) : (resa 1) veut réserver 2 places de catégorie 1  
 Spectateur 0 (3075832640) : (resa 1) obtient 2 places de catégorie 1  
 Spectateur 0 (3075832640) : (resa 1) occupe 2 places de catégorie 1  
 Spectateur 1 (3067439936) : (resa 1) veut réserver 2 places de catégorie 0  
 Spectateur 1 (3067439936) : (resa 1) obtient 2 places de catégorie 0  
 Spectateur 1 (3067439936) : (resa 1) occupe 2 places de catégorie 0  
 Spectateur 2 (3059047232) : (resa 1) veut réserver 2 places de catégorie 2  
 Spectateur 2 (3059047232) : (resa 1) obtient 2 places de catégorie 2  
 Spectateur 2 (3059047232) : (resa 1) occupe 2 places de catégorie 2  
 Spectateur 0 (3075832640) : (resa 1) a libéré 2 places de catégorie 1  
 Spectateur 0 (3075832640) : (resa 2) veut réserver 2 places de catégorie 1  
 Spectateur 0 (3075832640) : (resa 2) obtient 2 places de catégorie 1  
 Spectateur 0 (3075832640) : (resa 2) occupe 2 places de catégorie 1  
 Spectateur 1 (3067439936) : (resa 1) a libéré 2 places de catégorie 0  
 Spectateur 1 (3067439936) : (resa 2) veut réserver 2 places de catégorie 2  
 Spectateur 1 (3067439936) : (resa 2) je passe 1er pour réserver 2 places de catég. 2  
 Spectateur 2 (3059047232) : (resa 1) a libéré 2 places de catégorie 2  
 Spectateur 1 (3067439936) : (resa 2) obtient 2 places de catégorie 2  
 Spectateur 1 (3067439936) : (resa 2) occupe 2 places de catégorie 2  
 Spectateur 2 (3059047232) : (resa 2) veut réserver 2 places de catégorie 2  
 Spectateur 2 (3059047232) : (resa 2) je passe 1er pour réserver 2 places de catég. 2  
 Spectateur 0 (3075832640) : (resa 2) a libéré 2 places de catégorie 1  
 Spectateur 1 (3067439936) : (resa 2) a libéré 2 places de catégorie 2  
 Spectateur 2 (3059047232) : (resa 2) obtient 2 places de catégorie 2  
 Spectateur 2 (3059047232) : (resa 2) occupe 2 places de catégorie 2  
 Spectateur 2 (3059047232) : (resa 2) a libéré 2 places de catégorie 2  
 Fin de l'application de gestion des places