

# Programmation Concurrente

## *Travail Pratique sur les sémaphores/mutex*

### But du projet

Le but de ce travail pratique est de modéliser une salle de jeux vidéos. Le propriétaire de la salle n'a qu'une seule console de jeu, mais du dernier cri. La salle de jeux comporte 8 sièges dans une salle d'attente et un seul joueur peut jouer à la console à la fois.

Notre modélisation simule d'une part,  $M$  joueurs amis, possédant une somme partagée à dépenser en jeux de 100 CHF et d'autre part le comportement de la console de jeu.

**Tant qu'il y a encore de l'argent dans la caisse commune, le comportement de chaque joueur est le suivant :**

- a. il se promène en ville pendant un certain temps (4 à 8 ms dans le simulateur) avant de se rendre à la salle de jeux pour y jouer.
- b. lors de son entrée dans la salle, le joueur :
  - Regarde si un siège de la salle d'attente est libre. Si oui, il s'y assied, sinon il attend qu'un siège se libère.
  - Dès qu'il est assis dans la salle d'attente, il paye tout de suite 1 CHF du pot commun en avance pour pouvoir jouer ensuite au jeu. S'il n'y a plus d'argent dans la caisse commune, il sort définitivement et ne reviendra plus jouer (=sortie de thread).
  - Dès que la console de jeu est libre, un joueur assis en salle d'attente s'y rend pour y jouer. **IMPORTANT:** il n'y a aucun altruisme parmi les joueurs : le premier qui voit la console libre s'y installe!
  - Le joueur démarre le jeu et attend sa fin (game over).
  - Lorsque le jeu est terminé, il retourne se promener (point a) et recommence le cycle.

**Le comportement de la console est le suivant :**

- a) Elle indique qu'elle est prête à être utilisée.
- b) Elle attend qu'un joueur démarre le jeu.
- c) Elle simule une durée de jeu par une attente de 2 à 4 ms et indique que la partie est terminée.
- d) Elle s'arrête définitivement lorsque la 100<sup>e</sup> partie se termine, sinon elle reprend au point a).

### Cahier des charges

Le cahier des charges à remplir est le suivant:

- Votre programme doit comporter  $M$  threads pour les joueurs et 1 thread qui gère la console.  $M (\geq 1)$  doit être entré en ligne de commande.
- Chaque joueur doit compter le nombre de parties totales qu'il a fait et **l'afficher** à lorsque tout l'argent est dépensé.
- **L'utilisation de barrières est proscrite pour ce TP**
- Toutes les attentes doivent être passives.
- Tous les threads doivent se terminer normalement quand la caisse commune des joueurs est vide.
- Veuillez **respecter** les directives décrites dans le document « Consignes pour les travaux pratiques » figurant en début de cours sur Cyberlearn.

- Lorsque votre code est terminé et testé, ôtez les délais imposés par cette spécification et vérifiez que votre code fonctionne encore.

## Rendu du TP

Constituez et rendez sur Cyberlearn une archive nommée *G<votre numéro de groupe>\_TP3.zip*.

Celle-ci doit contenir vos fichiers source et un makefile dont la cible ‘clean’ permet d’effacer l’exécutable et les fichiers temporaires liés à la compilation. La commande ‘make’ doit permettre de **compiler vos sources sans erreurs et sans warnings** (avec les options -Wall et -Wextra).

## Exemple d'exécution

Simulation with 20 players :

```
./tp3_salle_arcade 20
Simulation with 20 players and 8 waiting seats
[GAME] waiting player to start...
[player 1] strolls around the city...
[player 1] enters the arcade
[player 1] is in waiting room
[player 1] pays a coin. Remaining players cash: 99
[player 1] plays until game is over...
[GAME] start...
[GAME] finished (game over)! Machine cash: 1
[GAME] waiting player to start...
[player 1] Leaves the shop (nb_games: 1)
[player 1] strolls around the city...
[player 1] enters the arcade
[player 1] is in waiting room
[player 1] pays a coin. Remaining players cash: 98
[player 1] plays until game is over...
[player 2] strolls around the city...
[player 2] enters the arcade
[player 2] is in waiting room
[player 2] pays a coin. Remaining players cash: 97
[GAME] start...
[GAME] finished (game over)! Machine cash: 2
[GAME] waiting player to start...
[player 1] Leaves the shop (nb_games: 2)
[player 1] strolls around the city...
[player 1] enters the arcade
[player 1] is in waiting room
[player 1] pays a coin. Remaining players cash: 96
[player 4] strolls around the city...
[player 4] enters the arcade
[player 4] is in waiting room
[player 4] pays a coin. Remaining players cash: 95
[player 2] plays until game is over...
[GAME] start...
[GAME] finished (game over)! Machine cash: 3
```

...

```
[player 17] plays until game is over...
[GAME] start...
[GAME] finished (game over)! Machine cash: 99
[GAME] waiting player to start...
[player 7] plays until game is over...
[player 17] Leaves the shop (nb_games: 2)
[player 17] got 2 games at all
[GAME] start...
[GAME] finished (game over)! Machine cash: 100
[player 7] Leaves the shop (nb_games: 13)
[player 7] got 13 games at all
```