

TINAZZI Giovanni VANSON Nathan

RAPPORT TP7

Objectifs

L'objectif principal etait de creer deux programmes qui utilisent la memoire partagee pour accomplir une tache cooperative. Pour se faire, il fallait initialiser et utiliser une memoire partagee, mais aussi utiliser des semaphores pour eviter les conditions de courses (conflits lorsque 2 processus sont vus comme concurrents sur l'acces des donnees).

Ici, notre programme est un restaurant represente par un serveur et un cuisinier, ou la memoire partagee va etre l'etagere qui relie les 2 entites.

Cook.c

Ici, le cuisiner devait s'occuper de cuire des pizza a poser sur l'etagere. Si 3 pizza sont sur l'etagere, il a le droit de reposer.

Tout d'abord, le cuisiner devait creer un segment de memoire partagee. Pour se faire, on utilise les commandes suivantes

```
// on va créer un segment de memoire partagee
shm_fd = shm_open(name, O_CREAT | O_RDWR, 0666);

// on va configurer la taille du segment de memoire partagee
ftruncate(shm_fd, sizeof(int));

// on va mapper le segment de memoire partagee (etagere) dans l'espace
d'adresse du processus
shelf = mmap(0, sizeof(int), PROT_READ | PROT_WRITE, MAP_SHARED, shm_fd,
0);
```

Apres avoir creer la memoire partagee et avoir set les semaphores, le cuisinier peut commencer a cuisiner. Etant donnee qu'il doit cuisiner 10 pizza, on va creer une boucle `while` qui va decrementer a chaque fois que une pizza est prete. Dans cette boucle on a:

```
sleep(rand()%2+1);
```

qui vas nous permettre de cuisiner une pizza en un temps aleatoire

L'unique condition etant que on peut avoir max 3 pizza sur l'etagere, notre cuisinier va donc cuisiner ses 10 pizza et s'arreter que s'il arrive la condition dite ci-dessus.

Une fois que toutes les pizza sont faites, on veiller à fermer et unlink tous les semaphores ainsi que la memoire partagee

waiter.c

Dans cette partie, il n'y a pas grand chose a dire car le principe est le meme que le module cook.c (il n'y a presque pas de difference à part pour la memoire partagee qui doit juste etre ouverte et ne dois pas etre set)

Comment utiliser le programme

Ici, on aura besoin de 2 terminaux differents: un pour le serveur et un pour le cuisinier. Apres avoir make sur un des 2 terminaux, on a juste a ecrire dans chaque terminal

```
./waiter
```

```
./cook
```

et le programme va agir tout seul