

TD4 threads/mutex/conditions

```
pthread_cond_t cond1; //identifiant de la condition
pthread_cond_init(&cond1, NULL); //initialise une condition
pthread_cond_wait (&cond1, &mutex1); // attend le signal
pthread_cond_signal (&cond1); // envoie le signal
```

Exercice 1 : Les clients et le magasin possédant un seul article.

Ecrire un programme *c* qui crée *n* threads qui représentent les clients, puis un thread représentant le magasin. Toutes les secondes les clients vont acheter 1 à 10 unités de l'article. S'il n'y a plus d'article en magasin alors le client attend que le magasin se réapprovisionne. Toutes les 10 secondes le magasin remet le stock à 1000 unités.

L'article est représenté par une structure contenant simplement un entier correspondant aux nombres d'unités.

Exercice 2 : Clients pas content ...

Modifier le programme précédent de manière à ce que le client indique au magasin (à l'aide de signaux) qu'il n'y a plus d'article en rayons. Une fois le signal reçu, le magasin se dépêche de rajouter 100 unités de l'article en rayon, ce qui lui prend une seconde.

Exercice 3 : Plusieurs articles

Modifier le programme précédent pour qu'il y ait *m* articles (un tableau contenant la quantité de chaque article). Chaque client choisit de manière aléatoire l'article qu'il désire. Attention il vous faudra considérer un vendeur par article.

Exercice 4 : Plusieurs magasins

Ajouter plusieurs magasins (vous aurez une matrice où chaque ligne représente un magasin). Désormais le client choisit de manière aléatoire l'article et le magasin.

Exercice 5 : Le prix le moins cher ...

Chaque magasin possède un prix pour chaque article (il vous faudra une matrice qui donnera le nombre de chaque article et une seconde matrice avec le tarif de chaque article).

Le magasin met à jour ses tarifs toutes les 3 secondes (ou quand il ajoute du stock sur un article). Le tarif de chaque produit est inversement proportionnel à la quantité restante, c.à.d. $\text{prix} = 10000/\text{quantité restante}$.

Bien évidemment, le client choisit le tarif le plus bas pour l'article désiré.

Exercice 6 : La pub

Chaque magasin fait un coup de pub toutes les 6 secondes, il choisit un article aléatoirement et s'aligne sur le prix le plus bas.

Exercice 7 : Aller plus loin

Ajouter un entrepôt, tous les magasins vont se servir dans le même entrepôt. Si l'entrepôt ne possède pas l'article souhaité alors le magasin s'excuse envers le client et le client repart sans l'article. De plus, les prix sont fixés par rapport au stock de l'entrepôt au moment où le magasin se sert dans l'entrepôt.