

TP PARALLELISME no 2

Synchronisation PThreads

Sujet: Synchronisation pour un affichage vidéo

Dans ce TP, on se propose de réaliser une synchronisation de type producteur consommateur mais où l'on ne bloque jamais le producteur quitte à perdre des données.

Le Thread « producteur », produit des *trames vidéo* avec une vitesse que l'on ne peut maîtriser et fait régulièrement appel à une fonction `void ecrire(int t)` qui doit recopier une trame vidéo dans le tampon.

Les trames sont numérotées. Pour simplifier on considèrera seulement qu'une trame est un entier.

Le Thread « consommateur » lit ces *trames vidéo* grâce à une fonction `int lire()` pour les afficher en prenant toujours la plus ancienne trame non encore lue.

En cas de saturation du tampon, le producteur remplace la trame la plus ancienne par une nouvelle trame.

Le producteur ne doit pas écraser la trame en cours de lecture ni se bloquer en attendant la fin de la lecture courante. Si il n'y a pas de trame non lue disponible le consommateur doit se bloquer.

Pour réaliser cette synchronisation on utilisera les wait/signal des Threads Posix. On pourra utiliser les structures de données suivantes pour gérer le tampon :

```
int etat[N]          /* etat[i] donne l'état de la trame qui est à l'emplacement i
                     dans le tampon =
                     1 : en_cours_de_lecture,
                     2 : lue ou vide
                     3 : non_lue,
                     4 : en_cours_d'écriture, */
int trame[N]         /* les trames */
```

Il peut également être utile de découper la fonction `lire` en `debut_lire` - `_lire` - `fin_lire`. `debut_lire` doit rechercher la plus ancienne trame non lue et mettre à jour l'état de la trame, si aucune trame n'est disponible `debut_lire` doit être bloquante. `_lire` effectue la lecture et l'affichage de cette trame (à l'extérieur du moniteur !). `fin_lire` est utilisée pour la mise à jour de l'état de la trame.

Question 1 : écrire le programme avec les threads producteurs et consommateurs et un tampon géré de manière classique (blocage éventuel de l'émetteur)

Question 2 : remplacer la mise en oeuvre du tampon par le tampon qui ne bloque pas le producteur

Question 3 : tester la mise en oeuvre. Mettre en évidence les différents cas possibles.

Question 4 : modifier le tampon pour qu'il puisse gérer deux "consommateurs". Dans ce cas la consommation n'est effective que lorsque les 2 consommateurs ont effectivement affiché la trame.

Question subsidiaire : peut on généraliser à un nombre inconnu de consommateurs ?
