Sujet de TP 3 **Système** et réseau - Programmation système Licence 3

Synchronisation

V. Felea & A. Hugeat

Les exercices suivants nécessitent l'utilisation des outils de synchronisation de l'accès en exclusion mutuelle à une ressource partagée. Ce partage peut être réalisé entre processus par segments de mémoire partagée. Quant aux threads, le partage de données peut être réalisé grâce au tas ou au segment de données (partagé par défaut entre les threads d'un même processus). Deux techniques d'échange d'information entre threads sont possibles : les variables globales et le passage des paramètres.
Remarque De manière générale, la synchronisation ne se limite pas à l'implémentation de l'exclusion mutuelle.

1. **Somme des éléments d'une matrice avec synchronisation**
   Reprendre l'exercice sur la somme des éléments d'une matrice d'entiers. Écrire une version dans laquelle chaque thread rajoute son résultat (la somme des éléments de la ligne concernée) à la somme finale, qui est une variable entière partagée par tous les threads. La synchronisation s'appuiera sur des mutex.

2. **Somme des pairs et somme des impairs**
   Deux processus, en relation père-fils, exécutent le même procédé : ils génèrent n valeurs entières dont les pairs sont rajoutés à une variable partagée `sPairs` et les impairs sont rajoutés à une variable partagée `sImpairs`. Une fois la génération des nombres terminée et les additions effectuées, le père doit afficher les valeurs des deux variables `sPairs` et `sImpairs`.

3. **Somme des éléments d'une matrice avec synchronisation - bis**

   Reprendre l'exercice 1, en réalisant la synchronisation des threads grâce aux sémaphores POSIX.
   Remarque Si l'implémentation de l'exercice 1 utilise des variables globales, utiliser pour cette solution le passage de paramètres et inversement.

4. **Algorithme de Peterson**

   Une solution au problème de l'exclusion mutuelle entre deux processus `p0` et `p1` par variables partagées est due à G.I. Peterson (1981). Les algorithmes des processus autour des sections critiques sont les suivants, avec `tour`, `D0` et `D1` trois variables booléennes partagées (des entiers) :

```
Processus p0                    Processus p1
------------                    ------------
/* avant SC */                  /* avant SC */
D0   = 1                        D1   = 1
tour = 0                        tour = 1
tantque D1 et tour == 0         tantque D0 et tour == 1
fait                            fait
/* SC */                        /* SC */
D0 = 0                          D1 = 0
/* après SC */                  /* après SC */
```

Implémenter un tel mécanisme à l'aide des segments de mémoire partagée. L'initialisation de `D0` et `D1` est à `faux`, celle de `tour` est à 0.