

# Master Ingénierie du Logiciel Libre – 2<sup>ème</sup> Année (M2I2L)

## Module système, réseaux et sécurité

### Année 2009-2010

### C.R. TP N°5 : Lecteurs-Rédacteurs

D. DUVIVIER ([duvivier@lil.univ-littoral.fr](mailto:duvivier@lil.univ-littoral.fr)) – LIL/ULCO

---

Il s'agit de réaliser une synchronisation entre plusieurs processus lecteurs et plusieurs processus rédacteurs en utilisant le modèle « lecteurs/rédacteurs » pour synchroniser les accès à un objet partagé entre l'ensemble des processus. Cet objet peut être assimilé à un fichier. Dans le programme « C » que vous me remettrez en guise de compte-rendu, il n'est pas nécessaire d'implémenter effectivement cet objet partagé. Vous afficherez simplement des messages pour simuler les accès +/- concurrents en lecture et/ou en écriture. De même, toujours pour simplifier l'écriture du programme, la variable partagée « NLC », destinée à compter le nombre de lecteurs en cours d'accès, pourra être avantageusement remplacée par un sémaphore. Ceci permet de remplacer la mise en place d'une variable entière « NLC » dans un segment mémoire partagé, par l'utilisation de la valeur (positive ou nulle) d'un sémaphore « NLC ». Pour incrémenter NLC, il suffit d'utiliser V(NLC), pour la décrémenter P(NLC). Dans les pseudo-code présentés ci-dessous, les temporisations (via l'appel système sleep()) n'apparaissent pas, mais il est évident qu'il faut prévoir des temporisations dans et or section(s) critique(s). Prévoyez aussi un code qui s'arrête « proprement » après un certain nombre d'itérations en libérant correctement les ressources système. Pour lancer les rédacteurs et les producteurs ainsi que pour utiliser les sémaphores, basez-vous sur les codes mis à votre disposition en TP.

Le compte-rendu comportera les fichiers source (\*.c et \*.h) ainsi que le fichier makefile s'il s'avère nécessaire. N'oubliez pas de faire apparaître vos noms et prénoms dans chaque fichier, n'oubliez pas de faire figurer le nom de votre binôme si vous travaillez en binôme bien sûr !

Initialisation :
------------------

- Mutex = 1
- Fichier\_Libre = 1
- NLC = 0

Pseudo-code pour le(s) rédacteur(s) :
---------------------------------------

```
tant-que (Vrai) {
    /* Demander écriture */
    P(Fichier_Libre)

    /* Ecrire dans le fichier */
    ...

    /* Signaler fin d'écriture */
    V(Fichier_Libre)
}
fin-tant-que
```

Pseudo-code pour le(s) lecteur(s) :
-------------------------------------

```
tant-que (Vrai) {
    /* Demander lecture */
    P(Mutex)
    si (NLC == 0) alors
        P(Fichier_Libre)
    fin-si
    NLC = NLC + 1
    V(Mutex)

    /* Lire dans le fichier */
    ...

    /* Signaler fin de lecture */
    P(Mutex)
    NLC = NLC - 1
    si (NLC == 0) alors
        V(Fichier_Libre)
    fin-si
    V(Mutex)
fin-tant-que
```