

## TD 2 – ASR7 Programmation Concurrente

Thread et concurrence

8 octobre 2018

### I Retour sur le TD1

- Exercices 2 et 4 pour le groupe A
- Exercices 3 et 4 pour le groupe B

### II Pourquoi faire un programme multithread

On souhaite comparer l'efficacité d'un serveur de fichiers en mode mono ou multithread, même sur un ordinateur disposant d'un seul processeur monocoeur.

L'intérêt du multithread se trouve uniquement lors des accès disque car le thread qui demande l'accès à un fichier doit attendre pendant que les données sont lues sur le disque. Le serveur de fichiers dispose d'un cache en mémoire pour les fichiers les plus couramment lus. On suppose :

- la durée pour traiter une requête sans accès disque est de 15ms (récupérer la requête, chercher dans le cache, rendre le résultat) ;
- pour gérer le multithreading un surcoût de 5ms est nécessaire (changement de contexte et passage en mode noyau pour passer d'un thread à l'autre) ;
- si le fichier ne se trouve pas en cache il faut 75ms supplémentaires pour la lecture sur le disque ;
- en moyenne un fichier est disponible dans le cache dans 2/3 des cas.

- Q.II.1)** - Donner le nombre de requêtes traitées par seconde pour un serveur monothread
- Q.II.2)** - Donner le nombre de requêtes traitées par seconde pour un serveur multithread utilisant des threads noyaux.
- Q.II.3)** - Est-il intéressant d'utiliser des threads utilisateurs (green threads) ?