

UNIVERSITÉ DE SHERBROOKE
DÉPARTEMENT D'INFORMATIQUE

IFT 630

Processus concurrents et parallélisme

Devoir no 3

Devoir à remettre au plus tard le vendredi 3 août 2018.

La correction s'appuie en partie sur les instructions de déploiement et d'utilisation des programmes que vous devez inscrire dans votre rapport de travail pratique. La qualité du français écrit est de rigueur.

La charge de travail convient à une équipe de deux personnes. Annoncez cette composition aussitôt que possible.

Contraintes :

- Les programmes conçus avec les outils installés dans les laboratoires du département.
- Les programmes doivent s'exécuter sans nécessiter de paramétrisation à la ligne de commande.
- La livraison s'effectue sur Moodle.

Travail

Vous devez produire :

- Une application client-serveur avec une communication de type message.
- Un rapport de travail pratique décomposé en cinq sections :
 - La page titre avec l'intitulé du cours et l'identification de tous les membres de l'équipe.
 - La table des matières.
 - Le compte-rendu de votre démarche et de vos recherches.
 - La description du mode de déploiement (procédure de compilation, lancement des programmes).
 - Une liste de références (n'oubliez aucune référence).

Description des problèmes

Vous devez implanter un serveur en C++ en exploitant la concurrence offerte par les systèmes POSIX ou de la bibliothèque STL de C++. Vous devez établir un protocole de communication par messages entre les clients et le serveur. Vous restez libres d'utiliser un canal de type « pipe », un « mailslot » ou « socket ».

Votre serveur devra imiter un serveur de transfert de fichiers ou tout autre type de serveur susceptible de faire ressortir les avantages des multiples fils d'exécution (multi-threads). Pour mettre le parallélisme en relief, produisez plusieurs clients (processus et non des fils d'exécution) qui acheminent des demandes simultanées. Le serveur doit assurer un bon fonctionnement et se montrer fiable, c.-à-d. terminer proprement même en présence d'erreurs irrécupérables. Pour y parvenir, vous devez utiliser les « signaux » de POSIX et les capturer tous.

Pour le démarrage, vous devez utiliser un processus principal qui utilisera les commandes « fork » et « exec ». Ce processus démarre tous les clients ainsi que le serveur et synchronise la fin du système. Un des clients doit pouvoir démarrer en mode interactif pour vérifier la bonne capture des signaux.