# Programmation système II : les threads

Timothée Ravier, LIFO, INSA-CVL, LIPN

1re année cycle ingénieur STI, 2013 – 2014

Les codes sources de ce TD sont disponibles à l'adresse : https://tim.siosm.fr/cours/.

## 1 Comparaison processus vs threads

Lisez, compilez et testez les temps d'exécution des programmes `0_process.c` et `0_thread.c`.

- **Question 1 :** Expliquez les différences en temps d'exécution et en usage du système.
- **Question 2 :** Comparez vos résultats avec les chiffres en [8].

## 2 Comparaison séquentiel vs parallèle

Lisez, compilez et testez le temps d'exécution du programme `0_multicoretest.c`.

- **Question 3 :** Expliquez les différences en temps d'exécution et en usage du système.

## 3 Création de threads et accès concurrents

Lisez, notez ce que vous comprenez du comportement du programme, compilez et testez le programme `1_create.c`.

- **Question 4 :** Expliquez le comportement obtenu. Pourquoi est-ce que l'on affiche des valeurs supérieures à 40 ?
- **Question 5 :** Pourquoi est-ce que l'on ne voit jamais la boucle principale (main) afficher une valeur de compteur supérieure à 0 ? Pourquoi voit-on presque exclusivement le thread 0 ?
- **Question 6 :** Modifiez le programme pour que tous les threads aient une chance.
- **Question 7 :** Proposez une correction de ce programme. Quel mécanisme pourrait-on utiliser pour s'assurer que la valeur affichée ne dépasse jamais 40 ?

## 4 Code de retour

Lisez, notez ce que vous comprenez du comportement du programme, compilez et testez le programme `2_join.c`.

- **Question 8 :** Essayez de donner le nombre -1 en entrée au programme. Que se passe-t-il ? Quel est le problème dans ce programme ? Proposez une solution.

## 5 Produit matriciel

- **Question 9 :** Écrivez un programme qui calcule le produit matriciel d'une matrice de très grande taille, sans thread.
- **Question 10 :** Modifiez votre programme pour répartir le calcul entre plusieurs threads. Faites varier le nombre de threads et constatez les variations en performance.

## 6 Récrire tee en version multi-thread

- **Question 11 :** Réécrire une version de `tee` qui lit un caractère après l'autre sur l'entrée standard et qui lance des threads pour les écrire dans deux fichiers différents. Faire une première version très naïve qui crée un thread pour chaque caractère lu.
- **Question 12 :** Écrire une version qui écrit un caractère sur deux dans des fichiers différents.
- **Question 13 :** Optimisez un peu les performances de votre programme en lisant plus d'un caractère à la fois et en utilisant des pipes par exemple.

## 7 Écrire un service « echo » en version multi-thread

- **Question 14 :** Inspirez-vous de vos cours de réseau [8] pour écrire un programme capable de répondre à plusieurs connexions simultanément en utilisant des threads.

## 8 Références


— POSIX Threads Programming : https://computing.llnl.gov/tutorials/pthreads/#WhyPthreads
— Multicore programming with pthreads : http://ashishagarwal.org/2011/02/13/multicore-programming-with-pthreads/
— Beej's Guide to Network Programming : http://beej.us/guide/bgnet/