

# INF4033 Lab 6 :

## Pour aller plus loin avec les Pthreads

Alexandre BRIÈRE

## 1 Introduction

### 1.1 Objectifs

Mettre en pratique le paradigme "divide and conquer" grâce aux *Pthreads*.

Vous n'utiliserez pour ce TP que les fonctions présentes dans la bibliothèque *Pthread* (fork, sémaphore, etc. ne sont pas autorisés).

### 1.2 Pour commencer

Les codes sources sont fournis avec un makefile, tapez la commande suivante pour compiler :

```
$ make
```

### 1.3 Astuces

Pour connaître le prototype exact des fonctions de la bibliothèque Pthread, pensez à la commande `man`.

```
$ man pthread_create
```

Pensez bien à vérifier toutes les valeurs de retour de la bibliothèque Pthread.

## 2 Exercice

Le but de l'exercice est de réaliser un algorithme de tri parallèle. L'application de tri a de nombreuses ressemblances avec l'application recherche max. Chaque thread trie individuellement sa partie de tableau. La fonction `sort_partial` permet de faire ce travail. Une fois les sous-tableaux triés, les threads se regroupent deux à deux (*terminant* et *fusionnant*). Chaque thread *terminant* retourne son tableau trié au thread *fusionnant* associé. Chaque thread *fusionnant* attend la fin du thread *terminant* associé, puis commence à fusionner son tableau avec celui du thread attendu.

Pour la première itération, les threads *fusionnants* sont ceux dont l'index est multiple de deux, les autres threads étant *terminants*. Lors de la deuxième itération, les *fusionnants* sont multiples de quatre, les autres sont *terminants*. La troisième itération, les *fusionnants* sont multiples de huit, etc.

### 2.1 Completez le code

En vous servant du code de recherche du max et en utilisant le code donné dans le dossier `exo`, faites en sorte que le programme trie le tableau avec plusieurs threads.

Pour compiler le code, utilisez la commande `make`. Le programme attend en paramètre le nombre de threads à lancer pour effectuer le tri.

## **2.2 Mesure de performance**

Le programme écrit sur la sortie d'erreur le nombre de secondes passées à faire le tri. Tracez le temps d'exécution en fonction du nombre de threads.

Que constatez-vous ? Comment expliquez-vous cela ?