

### TP N°3-b : Processus & threads – 1exercice – Durée approx. ~ 0h45min.

A l'aide du support de cours et des éventuels mémentos réalisez l'exercice suivants :

(Tout code ou implémentation compilant ou non sera étudié)

Le présent sujet de TP comporte 2 pages

#### I – « Threaded sort »

Écrivez un programme qui initialise un grand tableau d'entiers avec des valeurs aléatoires, recherche ensuite le minimum et le maximum du tableau et affiche le résultat.

Le tableau sera déclaré en variable globale :

```
#define SIZE (int)1e8  
int tab[SIZE];
```

Le programme affichera également le temps de la recherche du **min** et du **max** (temps d'initialisation non compris).  
On utilisera la fonction **gettimeofday**.

- Proposez une version threadée de ce programme. Le résultat sera écrit dans une ou plusieurs variables globales (une seule variable si une structure est utilisée). Testez ce programme avec 2, 4, 8 threads et en augmentant la taille du tableau. Que constatez-vous ?
- Ajoutez ensuite un mutex pour protéger l'accès aux variables globales résultats.

Remarque : on utilise **gettimeofday** et non **clock** car gettimeofday rend le temps machine réel alors que clock donne la somme des temps d'exécution des threads (principal et fils).

## **IV – Rappels**

### **Approche incrémentale du développement**

Pour obtenir les résultats attendus aux différents exercices, veuillez toujours appliquer une approche incrémentale en termes d'ajout de code/fonctionnalité.

Par exemple: une approche incrémentale pour ce type d'exercice "Is-like" serait :

1. La récupération des paramètres.
2. Tester fichier/répertoire.
3. Parcourir les éléments du répertoire en affichant leur nom.
4. Alimenter chaque fichier avec une information supplémentaire : permission / taille / propriétaire...

### **Documentation**

Pour obtenir des informations ou de la documentation ayez le réflexe d'utiliser les pages du manuel.

Par exemple:

➤ man 3 stat / man 2 open / man 2 readdir / man errno

### **Gestion des erreurs**

Afin d'avoir une gestion des erreurs la plus précise possible ayez le réflexe d'utiliser les codes retours **ERRNO** spécifiés dans les pages de manuel

Par exemple:

- |                  |   |
|------------------|---|
| ➤ <b>EEXIST</b>  | File exists (POSIX.1)                       |
| ➤ <b>EFAULT</b>  | Bad address (POSIX.1)                       |
| ➤ <b>EISDIR</b>  | Is a directory (POSIX.1)                    |
| ➤ <b>ENOTDIR</b> | Not a directory (POSIX.1)                   |
| ➤ <b>ELOOP</b>   | Too many levels of symbolic links (POSIX.1) |