

## **M3101 : Principes des systèmes d'exploitations**

### **TDM5 : Synchronisation des threads sous Linux**

---

#### **Objectif :**

Mise en œuvre de la bibliothèque des threads du système **Linux**.

---

#### **Travail demandé :**

Dans le répertoire **TP5**, développer en langage **C**, les applications ci-dessous :

Tous les processus ou threads devront commencer par afficher sur la sortie standard, la date et l'heure puis le login et l'UID de l'utilisateur, avant de réaliser le traitement demandé.

#### **Exercice 1 :**

Soient plusieurs threads **concurrents** qui doivent se partager le terminal en **exclusion mutuelle** afin que les données affichées sur la sortie standard par chaque thread, ne soient pas **entrelacées** et que les données saisies sur l'entrée standard, soient lues par le thread auquel elles sont destinées. Pour **synchroniser** ces threads, on utilisera un **mutex** :

- à un instant donné, un seul des threads possède le droit d'accès à la ressource.
- si un thread veut réaliser une E/S sur le terminal et qu'il ne possède pas le droit d'accès, il demandera à verrouiller le mutex, puis il attendra de l'obtenir.
- chaque fois que le thread qui possède le mutex, aura terminé une E/S, il transmettra le droit d'accès aux autres threads en libérant le mutex.

Les différents threads (au moins 3) seront créés par le thread principal qui ensuite se mettra en attente de leur terminaison.

Le traitement de tous les threads sera identique : par exemple entrer au clavier une suite de valeurs numériques avec un marqueur de fin et afficher le total cumulé à l'écran.

Le programme **source** sera codé dans le fichier **mutex.c** et le programme **exécutable** sera codé dans le fichier **mutex**.

## **Exercice 2 :**

On désire calculer les moyennes partielles et la moyenne générale d'un ensemble de notes d'un groupe d'étudiants.

Le traitement sera conçu comme un ensemble **d'activités parallèles** de la façon suivante :

- le processus (thread principal) lira les notes dans un fichier et les stockera dans un tableau à 2 dimensions à raison d'un étudiant par ligne et d'une matière par colonne. La désignation du fichier sera transmise en paramètre du **main**.
- le thread principal créera ensuite autant de threads qu'il y a de colonnes, chacun devant calculer la moyenne des notes de la colonne qui lui aura été assignée.
- chaque thread rangera sa moyenne dans un tableau, à l'indice correspondant à la colonne qui lui a été assignée.
- chaque thread synchronisera l'accès au tableau avec le thread principal, par l'intermédiaire d'un mutex.
- lorsque tous les threads auront terminé leur calcul, le thread principal calculera la moyenne générale et affichera les moyennes partielles et la moyenne générale.

Pour l'exercice, le nombre de matières sera fixé à **4** et le nombre d'étudiants à **10**.

Le tableau sera initialisé à partir du fichier texte **notes.data** sur Moodle.

**Le programme source sera codé dans le fichier `synchro2.c` et le programme exécutable sera codé dans le fichier `synchro2`.**