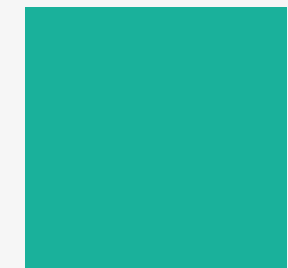


# **OS Multitâche**

## **Introduction avec Python**

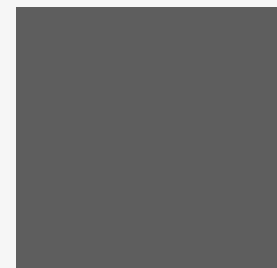


# CONTENU DU MODULE



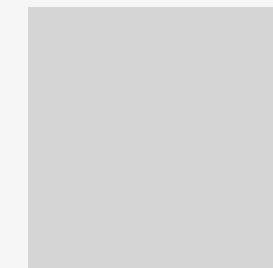
## TEMPS

- 16 heures.  
Pause pendant TP.



## OBJECTIFS

- Programmation concurrente
- Optimiser vos calculs



## NOTATION

- TP

# QU'EST-CE QUE LE MULTITHREADING ?

01



# 01

## LE MULTITHREADING

Quoi, pourquoi, comment ?



### Multithreading

Méthode de division d'exécution d'un programme

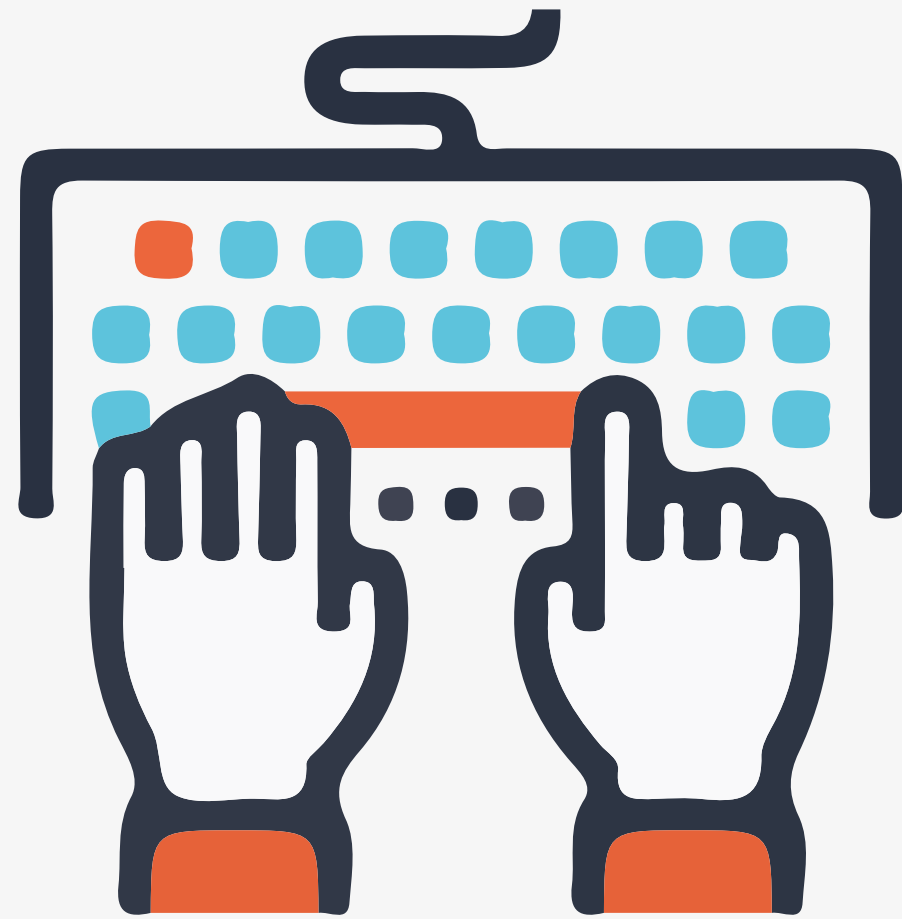
Permettre l'utilisation complète de ressources CPU

Un programme est divisé par des **Threads** et **Process**

# 01

## LE MULTITHREADING

Quoi, pourquoi, comment ?



### Programme

Un programme est un executable. Ex : run.exe, run.py ...

# 01

## LE MULTITHREADING

Quoi, pourquoi, comment ?



### ■ Processus

Un processus est une instance d'exécution d'un programme.  
Lancer un programme, ouvre un processus.

# 01

## LE MULTITHREADING

Quoi, pourquoi, comment ?



### Thread

- Un **thread** est une partie de l'exécution d'un processus. Un processus peut avoir un ou plusieurs threads.
- Il donne l'illusion du **multi-tasking**. Sans threads, un programme par tâche est nécessaire et les **processus** doivent être synchronisés.

**Ex :** IDE, navigateur ...

# 01

## LE MULTITHREADING

Quoi, pourquoi, comment ?



### Pourquoi utiliser les threads ?

Suivre l'avancée technologique

Création d'applications plus efficaces

Utiliser l'intégralité des ressources CPU

Les threads sont plus légers que les processus

Permettre le **multitâche**



# 01

## LE MULTITHREADING

Quoi, pourquoi, comment ?



### ■ Limitation des threads

■ Plus de threads ne veut pas toujours dire plus rapide

■ Plus de threads impliquent une gestion complexe

■ Pas d'utilité pour des opérations séquentielles

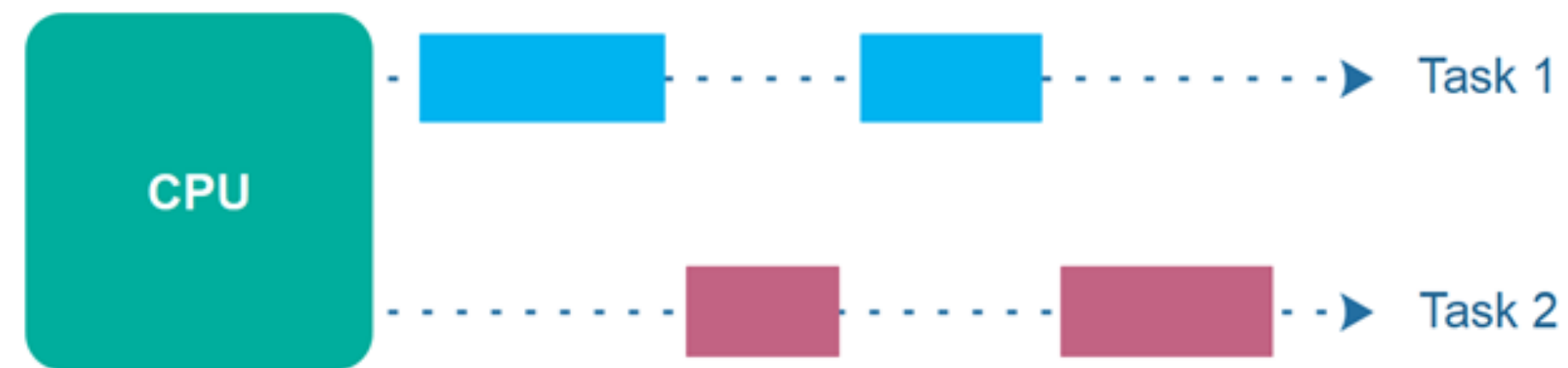
# CONCEPTS CLÉS

02 —

# 02

## LE MULTITHREADING

### Concepts clés



### La concurrence

Progression de plusieurs tâches concurrentes

Différent du **parallélisme**

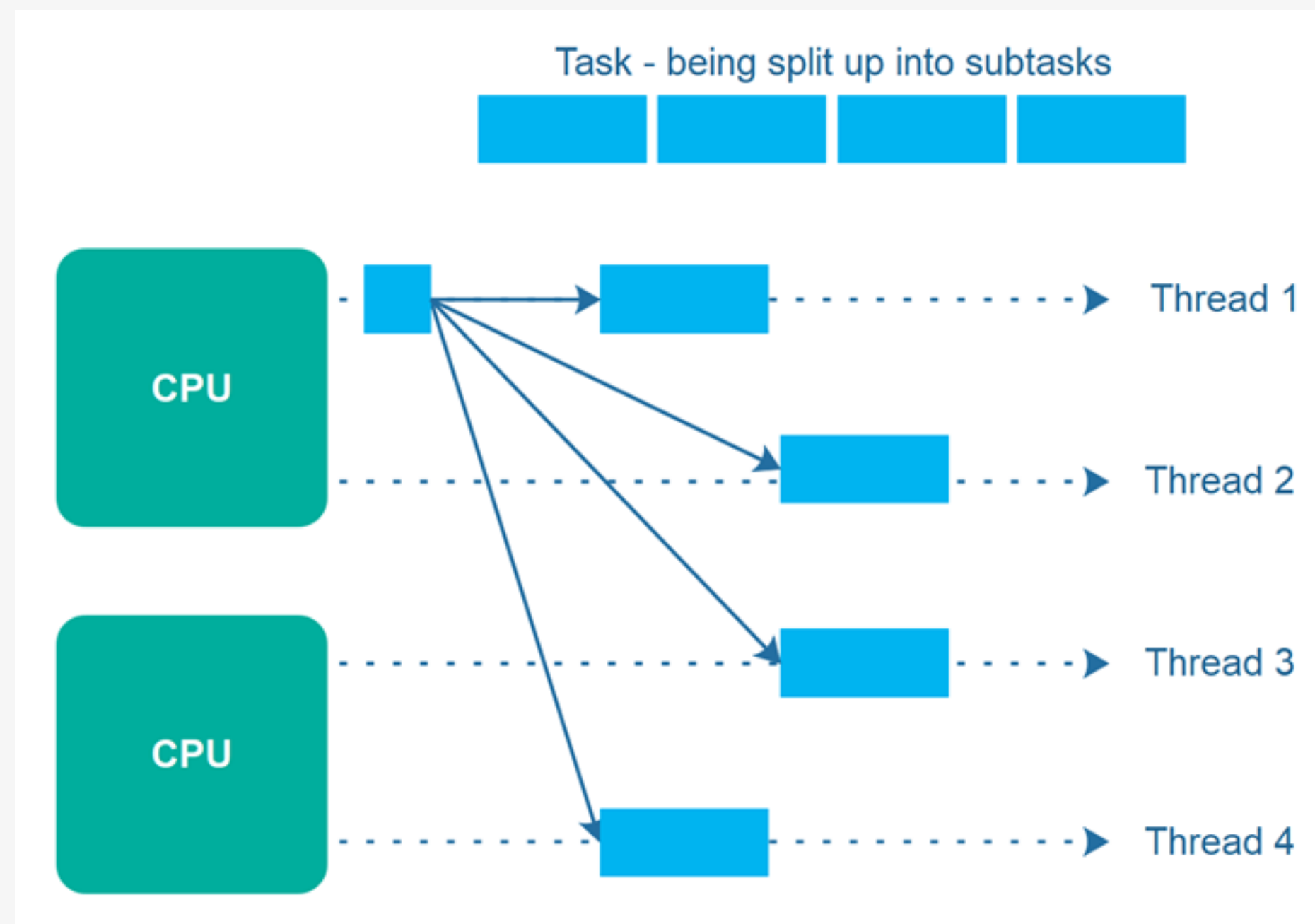
Le CPU **switch** entre les tâches

Possible par l'utilisation de threads

# 02

## LE MULTITHREADING

### Concepts clés



### Le parallélisme

Capacité de découpage de tâche

Exécution sur plusieurs CPU en même temps

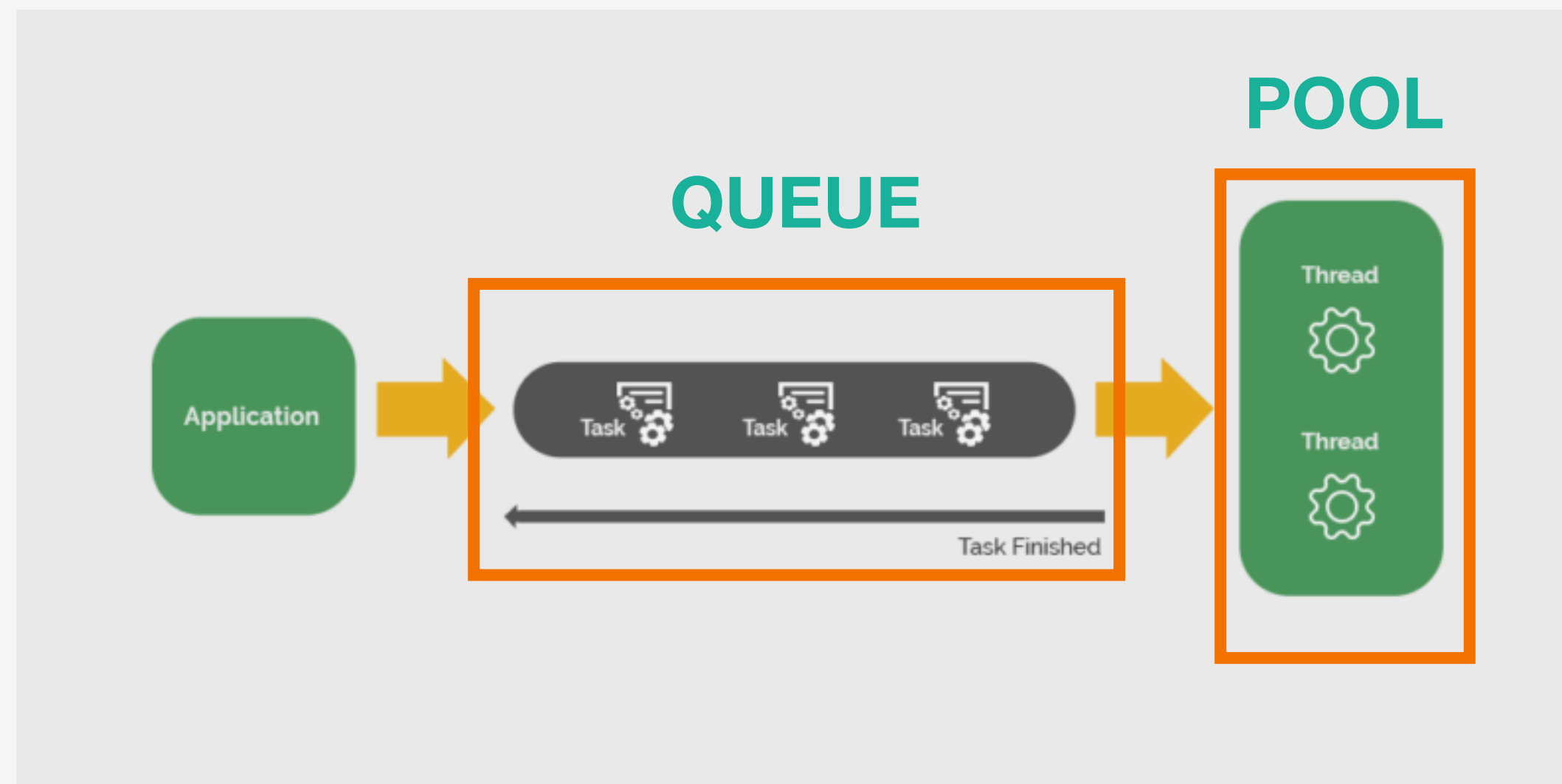
Possible avec plusieurs threads

Chaque thread exécuté sur un core différent

# 02

## LE MULTITHREADING

### Concepts clés



### Thread Pools

Découple la soumission et l'exécution de tâche

Affectation de thread

Thread retourne au Pool si la tâche se termine

Connecté à une **queue**

**Automatise** la création de threads

Les threads sont limités en mémoire

Si l'OS est chargé, l'application se dégrade proprement

# 02

## LE MULTITHREADING

### Concepts clés



#### ■ Locking

■ Technique de synchronisation

■ Limitation d'accès aux ressources

■ Permet le bon fonctionnement du multithreading

■ **Ex :** Mutex

# 02

## LE MULTITHREADING

### Concepts clés



#### Mutex

Exclusion mutuelle

Protection de données partagées

Permet l'accès à la ressource par un seul threads

**Ex :** Groupe de parole

# 02

## LE MULTITHREADING

### Concepts clés



### Semaphore

Mécanisme de signalement

Limite le nombre de threads sur une ressource partagée

Peut-être une stratégie de protection

**Ex :** Event / Groupe de parole



# PROBLÈMES IMPLIQUÉS

03



### Deadlock

Deux processus s'attendent

Blocage définitif

T1 veut la ressource de T2 et T2 celle de T1

**Ex :** Trafic à sens unique

# 03

## LE MULTITHREADING

### Problèmes impliqués



#### Race Condition

2 threads veulent modifier la même ressource en même temps

**Ex :** Circuit d'interrupteurs connectés

# 03

## LE MULTITHREADING

### Problèmes impliqués



#### Starvation

Un thread n'obtient pas l'accès à une ressource partagée

Un thread n'obtient pas de ressource CPU à cause d'autres threads

**Ex :** Restaurant universitaire



### Livelock

2 Threads agissent **en fonction de l'un et de l'autre**

**Arrêt** du progrès des threads

Forme de **starvation**

**Ex :** Personnes qui se croisent

# MULTITHREADING AVEC PYTHON

04



# 04

## LE MULTITHREADING

Avec Python

**TP1**



Écrire une fonction qui utilise N threads qui ajoute et supprime des éléments d'une liste. Montrer le mauvais fonctionnement du code.

**TP2**



Modifier la fonction pour permettre toutes opérations sur la liste par N threads de manière sécurisée.

**TP3**



Écrire une fonction qui permet à un thread de remplir une Queue. Si la Queue est pleine, un autre thread la vide. Si elle est vide le premier thread la remplit de nouveau. (Boucle infinie)

**TP4**



Modifier cette fonction pour permettre à N thread producteur et N thread consommateur.

**TP5**



Écrire un programme qui compte les lettres dans N fichiers texte par N threads. Le dernier thread affiche aussi le nombre total de lettres.

# 04

## LE MULTITHREADING

Avec Python

**TP6**



Messagerie instantanée (Thread / Event)

**TP7**



Website scrapping (Queues / ThreadPool)

**TP8**



Simulation achat / vente avec stocks (Locks)

**TP9**



Implémenter l'algorithme de Tri Pivot en utilisant des threads