

Les TPS "Projets"

CPE Année 2024-25

Les "projets" sont des applications plus conséquentes de la programmation concurrente avec Python.

Le site **e-campus** de ce module contient les dossiers "Cours Multiprocessing", "Projets Multiprocessing" et "Exemples Pour les Projets MultiProcessing".

Dans le répertoire "Projets Multiprocessing", deux fichiers .pdf contiennent les sujets proposés :

- Un premier "pdf" "1-Sujet-TP-CPE-Concurrent-24-25.pdf" contenant une première série d'exercices,
- Une seconde série d'exercices est également déposé dans le fichier "2-Sujet-TP-CPE-Concurrent-24-25.pdf"

Les codes de quelques exemples du cours ainsi que les codes fournis pour certains sujets de projet sont donnés dans le répertoire "Exemples Pour les Projets MultiProcessing".

Le principe des projets :

- Chaque exercice a un niveau de difficulté (de * à *****) avec un barème (de points).
 - Choisissez les sujets qui vous plaisent et réaliser un total de 20 points.
 - Vous pouvez réaliser ces projets en binôme (mais pas à 3 !). Si vous rendez votre travail par binôme, essayer de rendre > 20 points. Si un travail est rendu à 3 élèves, la note sera divisée par deux !
 - Les projets sont à réaliser sous Linux ou Mac, mais pas Windows (absence de fork) !
 - N'oubliez pas la partie "main" dans vos programmes Python. Elle est **obligatoire** pour le bon fonctionnement du package Multiprocessing que vous utilisez. Sa présence permet au correcteur de lancer simplement Python sur un fichier par projet (sans devoir appeler telle ou telle fonction à deviner) et de comparer les résultats obtenus avec la trace de l'exécution que vous avez rendu dans votre rapport pdf (lire ci-dessous).
- L'absence de la section "main" empêchera vos programmes de fonctionner correctement (et vous coutera un malus de la note). Lire à ce propos la documentation de Python.
- Si vous travaillez sous Mac, n'oubliez pas d'inclure le code du choix de "fork" au début du "main". Voir "2-Sujet-TP-CPE-Concurrent-24-25.pdf", page 2.
- Pompe.
- Vous ne devez pas utiliser le package "os" de Python pour un fork, l'utilisation d'une Queue, d'un Pipe, Théoriquement, le package Multiprocessing devrait vous suffir.

Les rendus

1- un fichier d'explication (pdf, docx, ...) contenant vos choix, vos explications, les principes de vos solutions) ainsi que les **traces d'exécution** des codes réalisés.

2- Les codes (séparément) permettant de lancer simplement "Python3" sur un fichier contenant le "main" principal du sujet traité.

N.B. : N'oubliez pas : si ChatGpt peut vous donner des solutions à certains de ces sujets, il est également capable de répondre à une question concernant vos rendus. Par exemple, "quelle est la source de << un morceau de votre code >>".

Le contenu du pdf "1-Sujet-TP-CPE-Concurrent-24-25.pdf" :

1. Course Hippique avec affichage à l'écran (non graphique)

Une version est fournie (le fichier " Course-hippique-eleves-avec-signal-pour-eleves.py" dans le répertoire " Exemples Pour les Projets MultiProcessing") où 20 processus sont créés et chaque processus représente un cheval. Testez-le.

Ce code contient tous les éléments pour les affichages, les codes d'échappement pour effacer l'écran, effacer une ligne, afficher en couleur, se placer à un endroit de l'écran, ...
Étudiez ce code puis ajoutez un **arbitre** qui affichera en permanence la position du gagnant.
Ne peut se réaliser sous Windows avec ces codes d'échappement fournis.

2. Client-serveur de calculs

Un / plusieurs serveurs de calculs + un ou plusieurs clients qui demandent à effectuer des calculs (comme " $2*5+6$ ").

Chaque client envoie une expression à évaluer dans une Queue, un des serveurs récupère une expression, l'évalue et renvoie le résultat.

3. Gestion des ressources (Billes)

Un gestionnaire de ressources (comme un gestionnaire de mémoire dans un système d'exploitation) reçoit des demandes de certaine quantité de ressources (ici ressources = billes) et essaie de satisfaire ces demandes.

4. Calcul parallèle d'estimation de Pi

Il y a différentes autres techniques de calcul de Pi.

Un exemple a été donné en cours 1 (calcul de Pi par la méthode arc-tg).

On demande à réaliser une approximation de la valeur de Pi par différentes méthodes proposées dans le sujet.

5. Calcul Clustering méthode K-means

Étant donné N points dans un plan 2D, regrouper ces points par proximité Euclidienne

6. Schéma Lecteurs / Rédacteurs

Simulation du travail d'un groupe d'élèves sur un projet partagé (avec lecture / modification simultanée des fichiers)

7. Exercices en annexes

7.1. Compléments sur le calcul de Pi

Cette partie propose d'autres méthodes d'estimation de la valeur de Pi.

7.2. Tri Merge-sort (tri fusion) en parallèle.

Le code de la version séquentielle de cette méthode est fourni. Il faudra comprendre cette méthode de tri puis la paralléliser en triant un grand tableau par plusieurs processus.

Il faudra démontrer le gain du temps par rapport à la méthode séquentielle.

7.3. Quick-sort (tri rapide) par de multiples Processus

Le code de la version séquentielle de cette méthode est fourni. Il faudra comprendre la méthode de tri puis la paralléliser pour trier un grand tableau par plusieurs processus.

Il faudra démontrer le gain du temps par rapport à la méthode séquentielle.

Les sujets du second pdf d'exercices (2-Sujet-TP-CPE-Concurrent-24-25.pdf)

Il contiendra les exercices suivants :

- La simulation d'un contrôleur de température / pression dans une salle,
- La simulation des activités d'un restaurant (clients, serveuses/serveurs, cuisiniers, ...)
- Le jeu de la vie
- Un simulateur de déplacements d'un Robot
- Etc

Bon travail !