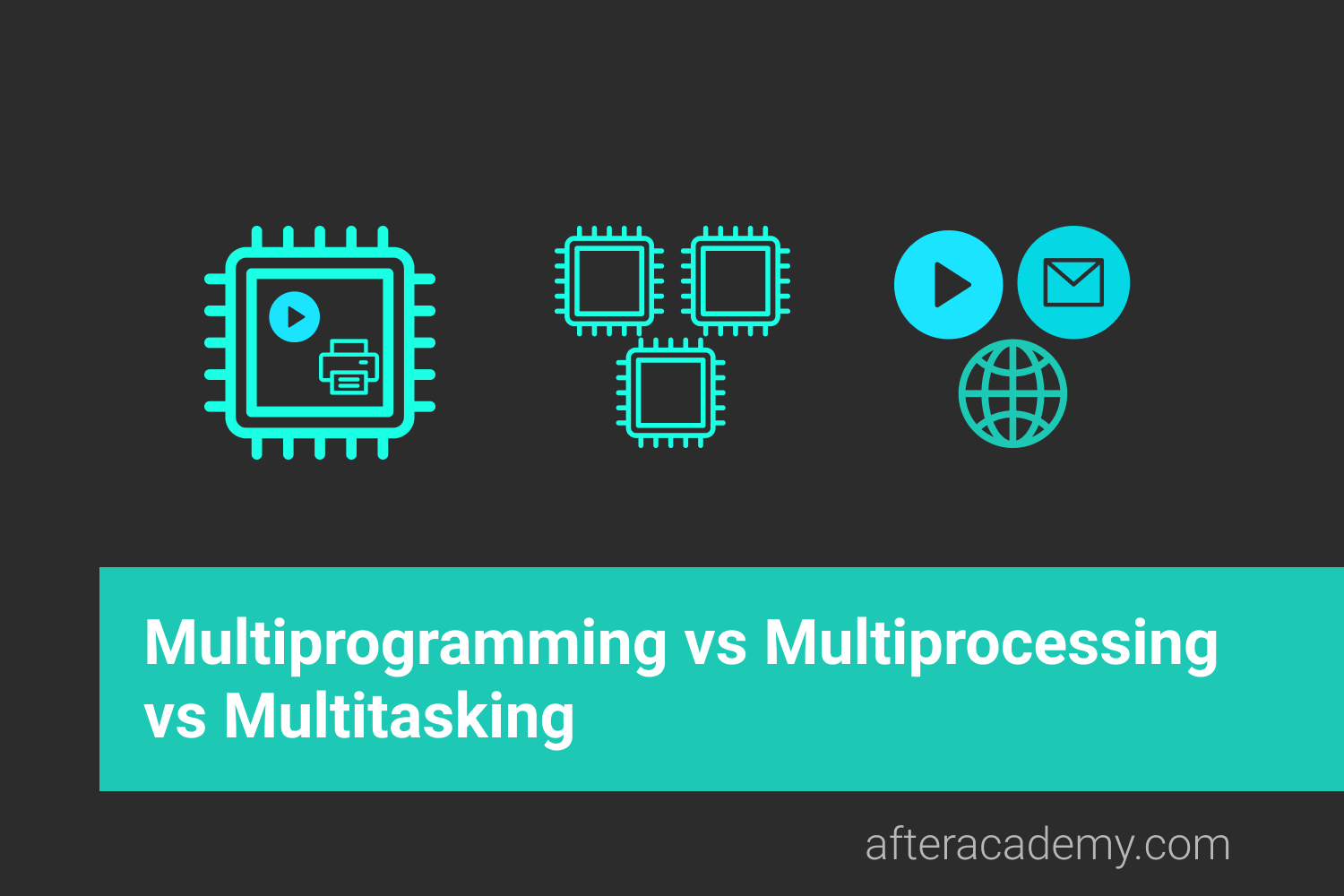
**Projet MultiProgramming en Python M1 :**

**Ajout d’un mécanisme de Multiprogramming**

**sur un serveur FTP distant**



PIERRE GIRAUD

CELESTIN CAPTAL

THOMAS BALDASSARRE

Contexte du projet :

Le projet consiste à l’ajout d’un mécanisme de multiprogramming tel que :

-*Du multi-threading*

-*Du multi-processing*

-*De la programmation asynchrone*

Il est donc demandé au groupe de choisir la meilleure solution selon le cas dans lequel il se trouve.Ce mécanisme sera alors ajouté à un programme qui réalise la synchronisation d’un dossier local d’une machine vers un site distant FTP . Toute modification sur dans le dossier local sera affecté sur le dossier FTP avec un délai d’attente.

Organisation :

Pour nous répartir le travail sur ce projet , nous avons crée un discord pour ce partager les différentes idéees et modifications que nous avions durant les cessions personnels .Thomas ayant eu des problèmes de compatibilité avec filezilla nous avons décidé de divisés le travail de la façon suivante :

* Pierre et Celestin se sont occupés de l’étude de fielzilla et de l’implémentation des fonctionnalités sur le programme.
* Thomas s’est occupé de l’analyse des différentes solutions en étudiant les TP réalisés précédemment.

Pour réaliser ce projet il est nécessaire d’étudier le code sur lequel nous allons ajouter la fonctionnalité de multiprogramming. Nous avons donc passé la première séance à l’étude de programme et à la compréhension de Filezilla.

Pendant ce temps Thomas a réfléchi aux solutions les plus optimales dans notre situation pour ne pas perdre de temps dans la réalisation du projet.

Les choix faits en terme du « **multiprogramming** » :

Pour savoir quels choix est le plus optimale pour améliorer le projet initial il était nécessaire d’étudier dans quelles parties le multiprogramming devra être effectué.

Multiple Connexion : Après avoir regarder le fonctionnement de Filezilla, nous avons pu comprendre qu’il était très compliqué de réaliser plusieurs manipulations sur le même dossier avec une seule connexion. Des interférences pouvant mettre à mal les opérations. Il est alors nécessaire de réaliser plusieurs connexions sur le même compte Filezilla. Nous voulons alors dans un premier temps réaliser plusieurs threads qui auront pour objectifs de gérer les multiples connexions sur le serveur. Il est possible de gérer le nombre de connexion sur Filezilla il n’est donc pas utile de créer une infinité de thread car le nombre de connexion simultané sera limité par le serveur. En effet nous avons décidé de partir sur du multithreading car les opérations que nous allons réalisés ne demandent pas trop de puissance il n’est donc pas nécessaire d’utiliser du multiprocessing qui requiert une plus grande rigueur dans son implémentation car il nous faudra plus de librairies et des ressources spécifiques. Cependant, le multithreading peut également être plus complexe à mettre en œuvre en raison de la nécessité de gérer correctement les problèmes de synchronisation. Mais dans notre cas il n’y aura pas un grand nombre de demandes simultanés ,nous n’aurons donc pas ce type de problème ou dans des situations très spécifiques .Le multiprocessing permet une meilleur isolation des taches ce qui augmentera la sécurité des interactions avec le serveur FTP .

En résumé, nous avons choisi l’optimisation des performances et la simplicité d’intégration du multithreading.

Difficultés rencontrées :

Nous avons rencontrées plusieurs difficultés pendant la réalisation de ce projet , cependant nous sommes tous en accord pour dire que celle qui nous a le plus mis en déroute est la compréhension de Filezilla et du FTP en général. En effet pour mettre en place un mécanisme de multiprogramming , il est nécessaire d’appréhender le fonctionnement du FTP. Les sources disponibles ayant souvent plusieurs années d’ancienneté. Cependant certains site comme Quora , Filezilla-project ou chatgpt pour les questionnement plus précis.