

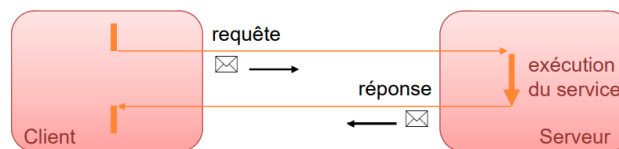
## TD noté : Programmation répartie

### Objectifs :

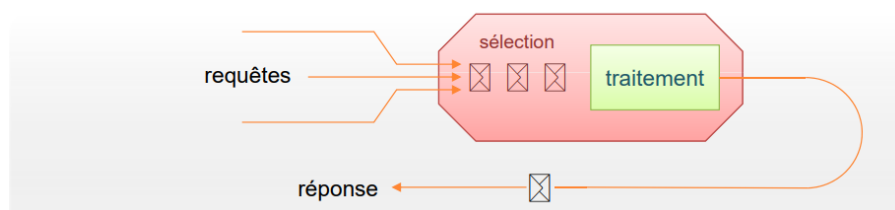
Utiliser les IPCs ou/et les threads pour réaliser une application type client serveur

L'objectif étant de créer une application type client/serveur. Cette application reposera donc sur le principe du client/serveur, c'est-à-dire une communication réalisée par dialogue entre deux processus : un processus serveur et un processus client.

- Le client initie l'échange ;
- Le serveur est à l'écoute d'une éventuelle requête d'un client ;
- Le client réceptionne les résultats délivrés par le serveur.



Le processus serveur est illustré par le schéma suivant :



- ✓ Reçoit la requête,
- ✓ Extrait le message,
- ✓ Réalise le traitement,
- ✓ Renvoi la réponse,

### Contexte et réalisation :

**Dans cette application vous n'allez développer ni un "vrai" serveur ni un "vrai" client qui peuvent dialoguer à distance, vous allez juste utiliser le principe client/serveur décrit ci-dessus.**

Le programme *Serveur* va permettre de calculer la moyenne des notes de la classe obtenue dans une matière donnée. Le *Client* envoie une requête au *Serveur* pour récupérer cette moyenne. Le *Serveur* renverra cette moyenne en réponse au client. **Chaque matière est caractérisée par un type, un nom et une moyenne. Le type étant un entier strictement positif.**

Le programme *Client* demande au serveur la moyenne d'une matière dont il va passer le type en paramètre sur la ligne de commande.

### Partie 1 :

*Dans cette partie, le programme Serveur se contente de répondre à un seul Client.*

1. Expliquez la solution choisie pour mettre en œuvre cette application :
  - ✓ Définition d'une matière, Créez un type **Matiere** permettant de correspondre à la description ci-dessus,
  - ✓ Choix de l'outil de communication,
  - ✓ Choix de (des) outil(s) de synchronisation,
2. Écrire le programme **Serveur** afin qu'il puisse récupérer le message d'un **Client** et y répondre.
3. Écrire le programme **Client** pour qu'il puisse transmettre une requête au **Serveur**, récupère le résultat et l'affiche. La requête devra préciser le type de la matière ; la réponse du serveur contiendra la moyenne de cette matière. *Le programme serveur devra donc déclarer et initialiser une structure de données contenant différentes matières avec le type et la moyenne correspondants. Par exemple un tableau défini comme ci-dessous :*

```
Matiere TabMatiere[] = {  
  { 1,"Anglais\0", 12.5},  
  { 2,"Mathematiques\0", 10.8},  
  { 3,"Informatique\0", 9.3},  
  { 4,"CulturteG\0", 12.6},  
  { 5,"Physiques\0", 11.8}  
};
```

### Partie 2 :

*Dans cette partie, le programme Serveur va devoir répondre à plusieurs programmes Client simultanément.*

1. Quelles modifications apportez-vous à la solution précédente ? justifiez vos choix ;
2. Écrire un programme **Serveur2** afin qu'il puisse mettre en œuvre les modifications ci-dessus et gérer la communication avec plusieurs programmes **Client**. Le programme **Serveur2** doit tourner en permanence.
3. Testez votre application en lançant plusieurs programmes **Client** simultanément.

### Travail à rendre :

1. Le code source complet des deux parties ; (Serveur.c, Sereveur2.c, Client et éventuellement un Client2 si vous modifiez le Client dans la seconde partie (**noté / 10**))
2. Un compte rendu précisant : (**noté / 10**)
  - a. Le but de cette application ;

- b. La solution mise en œuvre pour chacune des deux parties ;
- c. Les choix des outils utilisés pour la solution choisie (nombre d'outils, valeurs initiales pour certains outils, ...) et la justification de ces choix :
- ✓ Définir une matière ;
  - ✓ Calculer la moyenne ;
  - ✓ Initialiser un certain nombre de matières ;
  - ✓ La communication entre le serveur et un client ;
  - ✓ La synchronisation pour gérer le serveur et un seul client (Partie 1)
  - ✓ La synchronisation pour gérer le serveur et un seul client (Partie 2)
- d. La description des fonctions développées (type de retour, paramètres, ...) ;
- e. Une analyse des résultats obtenus ;
- f. Une conclusion sur le travail réalisé et une comparaison avec les autres outils que vous auriez pu utiliser ;

***Le travail sera à rendre au plus tard pour le 31/12/2022 à 22h***

Binôme N°	Etudiant1	Etudiant2	Binôme N°	Etudiant1	Etudiant2
1	LEROY Damien	MORIO Elias	17	BOUCHARD Hugo	CHERUEL Baptiste
2	LANGLOIS Gaëtan	LEBRASSEUR Thibaut	18	FEDORAWIEZ Thomas	
3	FOUQUET Victor	PEREIRA Théo	19	LESIEUR Romain	VASSE Samy
4	DEMELLIERS Léo	LECOUVREUR Romain	20	ORDONNAUD Mathieu	PLANTAIS Thomas
5	GONTIER-DURAND	PLISSONNIER Benoit	21	LECOUPANEC Hugo	MEREL Damien
6	LALLIER KEVIN	LAUNEY Samuel	22	GUILLOUET Adam	MEGARD Mylan
7	DENNETIERE Célia	DESQUESNES Sacha	23	LEBRETON Tais	LEBIEZ Théo
8	CAILLY Clément	TOLLEMER Hélie	24	CAUCHON Mahé	MARTIN Hugo
9	DESMONTS Arthur	GORLAS Faustine	25	BUKIN Augustin	GUERNINE Thanina
10	GOMOND Olivia	HAMEL Eva-Marie	26	LECROISEY Jean-Victor	LEVIEUX Tristan
11	PICHON Enola	QUERNE Iona	27	LEVEQUE Luc	TESSON Teiva
12	ANGOT Alexandre		28	MAFILLE Léo	POTTIER Eva
13	CUREAU Baptiste	HARNOIS Léo	29	GUILLOT Valentin	SEVAUX Clément
14	DUVAL ? Théo	EPPELE Ferdinand	30	CAILLY Bastien	COUEFFIN Louise
15	BELLAN Tristan	TOITOT Pierre	31	BRIZE Mael	
16	CLERICE Antoine	HELLUY Hugo			

***ANGOT Alexandre, FEDORAWIEZ Thomas et BRIZE Mael peuvent se mettre à deux ou faire le TP seuls.***