

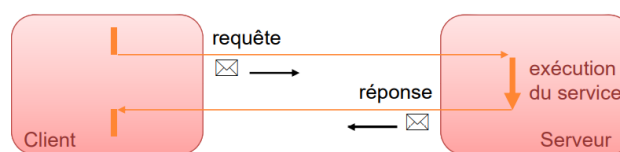
## TD noté : Programmation répartie

### Objectifs :

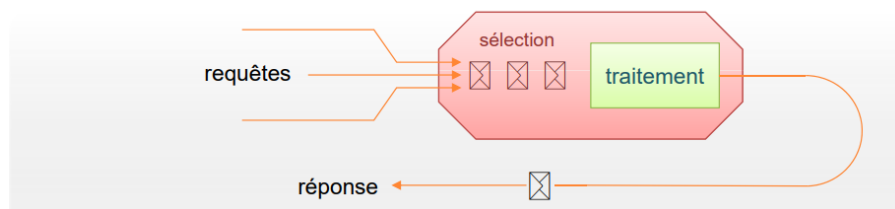
*Utiliser les sockets, les IPCs ou/et les threads pour réaliser une application client/serveur*

L'objectif de ce TP est de créer une application client/serveur TCP, 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 reçoit les résultats délivrés par le serveur.



Le processus serveur est illustré par le schéma suivant :



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

### Contexte et réalisation :

L'application à réaliser aura pour but de gérer les moyennes obtenues par une classe dans une matière donnée. Pour cela une solution est que Chaque matière soit caractérisée par un type, un nom et une moyenne. Le type étant un entier strictement positif.

Le programme *Serveur* initialisera un tableau avec les moyennes des matières enseignées. Le *Client* envoie une requête au *Serveur* pour récupérer cette moyenne. Le *Serveur* renverra la moyenne demandée en réponse au client.

Le *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 à la fois (Serveur mono 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 (par quelle outil le serveur et le client communiquent),
  - ✓ Choix de (des) outil(s) de synchronisation (par exemple sémaphore IPC, sémaphore POSIX ou mutex),

Vous pouvez faire un schéma qui illustre votre solution.

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**, récupérer 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 un tableau contenant différentes matières avec le type et la moyenne correspondants. Par exemple un tableau défini comme ci-dessous (où chaque matière est de Type **MATIERE**) :*

```
MATIERE TabMatiere[] = {  
{ 1,"Anglais\0", 12.5},  
{ 2,"Mathematiques\0", 10.8},  
{ 3,"Informatique\0", 9.3},  
{ 4,"Culturtre generale\0", 12.6},  
{ 5,"Physiques\0", 11.8},  
{ 6,"Programmation systeme\0", 12.3}  
};
```

## 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**.
3. Testez votre application en lançant plusieurs programmes **Client** simultanément.
4. Modifiez le programme **Serveur2** afin qu'il puisse s'arrêter s'il reçoit un message de fin.

## Travail à rendre :

1. Le code source complet des deux parties ; (Serveur.c, Sereveur2.c, Client.c et éventuellement un Client2.c si vous modifiez le Client dans la seconde partie (**noté / 10**))
2. Un compte rendu précisant : (**noté / 20**)
  - a. Le but de cette application ; (**1pt**)
  - b. La solution mise en œuvre pour chacune des deux parties ; (**2pts**)
  - 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 ; **(1pt)**
- ✓ La communication entre le serveur et un client ; **(3pts)**
- ✓ La synchronisation pour gérer le serveur et un seul client (Partie 1) ; **(1pt)**
- ✓ La synchronisation pour gérer le serveur et plusieurs clients (Partie 2) ; **(1pt)**

- d. La description des fonctions développées (type de retour, paramètres, ...) ; **(6pts)**
- e. Une analyse des résultats obtenus ; **(3pts)**
- f. Une conclusion sur le travail réalisé et une comparaison avec les autres outils que vous auriez pu utiliser ; **(2pt)**

***Le travail sera à rendre au plus tard pour le 31/12/2023 à 22h***

Binôme N°	Etudiant1	Etudiant2	Binôme N°	Etudiant1	Etudiant2
<b>1</b>	EPPELÉ Ferdinand	LEPLEY Gaëtan	<b>16</b>	DELCOURT Arthur	SNITH Valentin
<b>2</b>	BAYON Axel	LEVALLOIS Mathéo	<b>17</b>	SEFRIOUI Hugo	MOUCHEL Matthéo
<b>3</b>	LOUVET Flavien	MEZIERE Lény	<b>18</b>	MARGO Lisa	AUBRY Raphaël
<b>4</b>	PEYREGNE Nicolas	PETRUS Eliot	<b>19</b>	SOREL Lény	
<b>5</b>	LEROY Fabien	HAMON Arthur	<b>20</b>	LOPES Alenso	AÏT AZZOUZENE Julien
<b>6</b>	BAGOT Paul	BRINDJONC Lukian	<b>21</b>	DUBOS Arthur	CUREAU Baptiste
<b>7</b>	LE BLAY Ianis		<b>22</b>	TOURBILLON Noé	SAINT-GAUDIN Guilhem
<b>8</b>	RIGUET Cyrille	LAIGLE Émeline	<b>23</b>	LAGAÛZÈRE Hugo	DE LA POËZE D HARAMBURE Cyprien
<b>9</b>	JOBARD Maxime	SCHNEIDER Clément	<b>24</b>	LECOQ Enzo	LEVEQUE Luc
<b>10</b>	PIEL Louis	REKAÏ Vladimir	<b>25</b>	BRISAUD Andgel	FRIBOULET Victor
<b>11</b>	SIOPATHIS Lukas	LESIEUR Théo	<b>26</b>	LEVESQUE Maxence	BARATIN Clément
<b>12</b>	MADELAINE Baptiste	LE BLAIS Tom	<b>27</b>	SAILLY Julien	BALLU Martin
<b>13</b>	NICOLLE Raphaël	RICOZZI Romain	<b>28</b>	MORISSE Paul	BERGEROT Guillaume
<b>14</b>	LE NEINDRE Ethan	PATEY Valentin	<b>29</b>	ESSIENTH Orianne	HARANG Gabrielle
<b>15</b>	GUIDEAU Jean	LEROSIER Alexandre			

- ***Le tirage au sort a été fait de façon aléatoire sur plouf plouf en respectant les groupes de TDs***
- ***Mrs LE BLAY Ianis et SOREL Lény peuvent se mettre en binôme s'ils le souhaitent.***
- ***Les binômes se chargent de se répartir le travail ou pas, la note est attribuée aux deux, il vous revient donc de vérifier les travaux rendus.***