

Laboratoire 1
Application de bavardage



uOttawa

CEG 3585 - Comm données et réseautage

Université d'Ottawa

Professeur : Mohamed Ali Ibrahim

Noms et numéros des étudiants :
Gbegbe Decaho Jacques 300094197
Sissoko Ramatoullaye Bahio 300144949

Date de soumission:03-02-2023

Table des matières:

1. Description du problème
2. Explication du programme de solution
3. Document Représentatif (UML)
4. Captures d'écran
5. Discussion
6. Conclusion

1- Description du problème

Le but de ce laboratoire est de créer un programme de discussion clients serveurs axé sur système axé sur une connexion TCP/IP en utilisant des sockets.

Ce programme sera capable de lier plusieurs clients entre eux à travers un serveur commun sur lequel ils se seront connectés, le client pourra envoyer un message directement à un autre en privé s'il le veut.

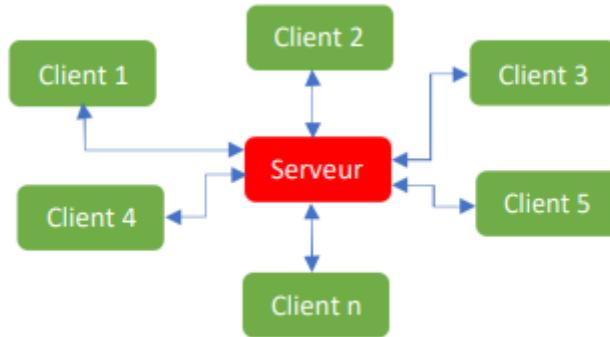


Figure 1 : Interconnexion entre le serveur et les clients

2- Explication du programme de solution

Pour pouvoir exécuter correctement le programme, il faut compiler le serveur en premier et le client ensuite dans un nouvel invité de commande.

Au lancement du programme client, on sera appelé à entrer notre identifiant. Dans le même temps, dans le serveur un message sera affiché comme quoi nous sommes connectés au serveur. Puis tous les messages envoyés sans précisions du destinataire seront affichés chez tous les participants du serveur ainsi que dans la file d'actualité du serveur.

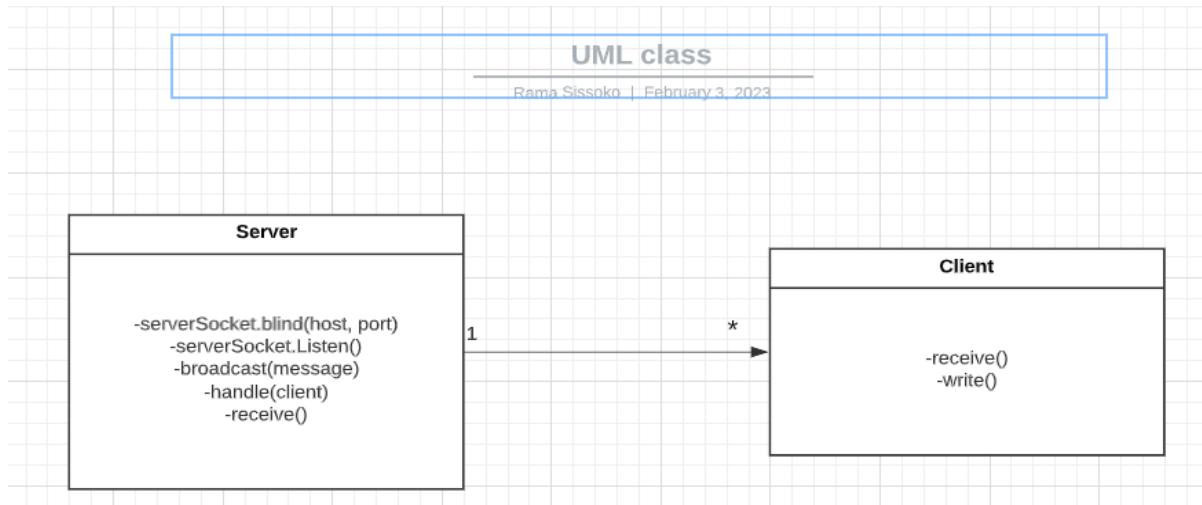
Lorsque un des participants précise un nom dans son message seul lui et cette personne verront le message et pourrons alors chatter ensemble jusqu'à ce qu'il sorte du chat.

Grâce à certaines fonctions des classes Clients.py et Server.py, la conception de programme à pu être réalisée. Ce sont en sommes :

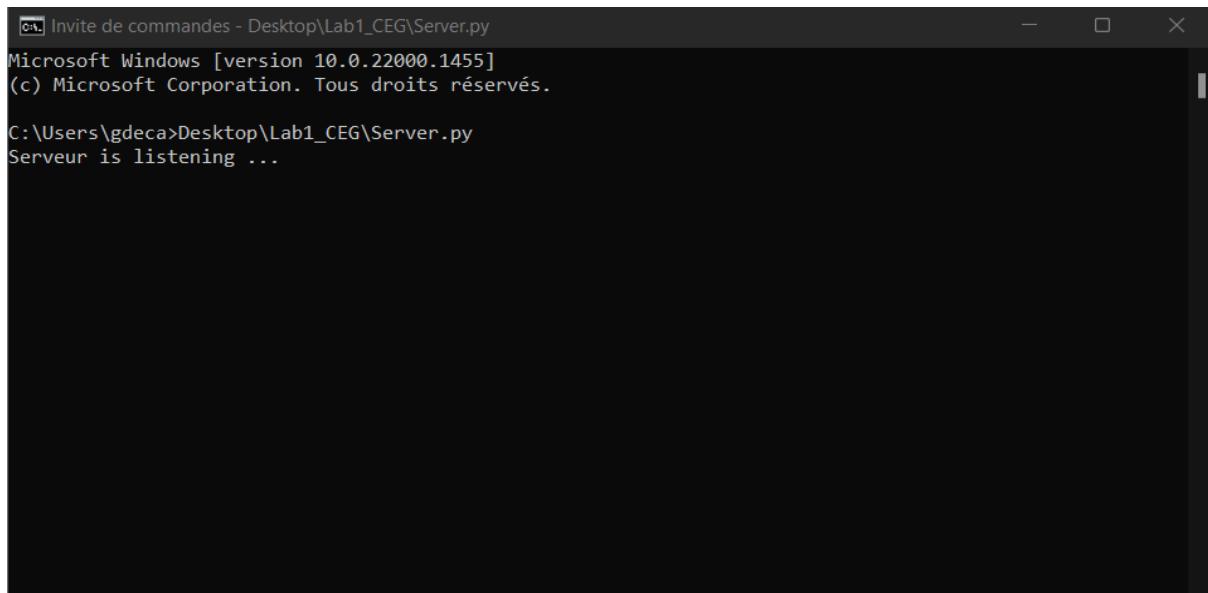
Classe Server.py		Classe Client.py	
Méthode	Explication	Méthode	Explication
socket.bind()	Elle permet de lier l'adresse (host et port) au socket	receive()	Elle retourne un paquet contenant toutes les infos de l'expéditeur
socket.listen()	Elle permet d'écouter essentiellement les connexions entrante sur le	write()	Elle associe le message à l'Id du client

	socket et spécifie le nombre max des connexions dans la file d'attente dans le Backlog		
broadcast()	Elle permet de diffuser un message sur une interface et de l'écouter sur une autre		
handle(client)	Elle permet de gérer les évènements liés au client		
receive()	Elle va afficher et spécifier qui envoie quoi à qui		

3- Diagramme UML



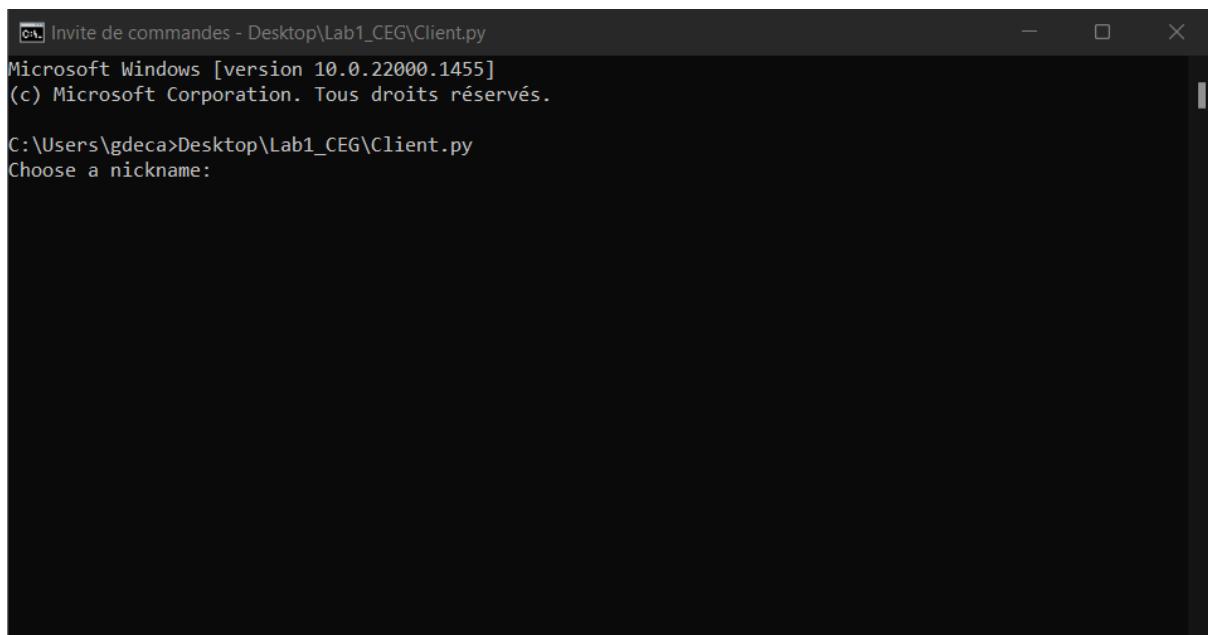
4- Captures d'écran



```
Invite de commandes - Desktop\Lab1_CEG\Server.py
Microsoft Windows [version 10.0.22000.1455]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\gdeca>Desktop\Lab1_CEG\Server.py
Serveur is listening ...
```

Connexion au serveur



```
Invite de commandes - Desktop\Lab1_CEG\Client.py
Microsoft Windows [version 10.0.22000.1455]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\gdeca>Desktop\Lab1_CEG\Client.py
Choose a nickname:
```

Connexion du client

```

[...] Invite de commandes - Desktop\Lab1_CEG\Server.py
Microsoft Windows [version 10.0.22000.1455]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\gdeca>Desktop\Lab1_CEG\Server.py
Serveur is listening...
Connected with('127.0.0.1', 51432)
Nickname of the client is Decaho!

```



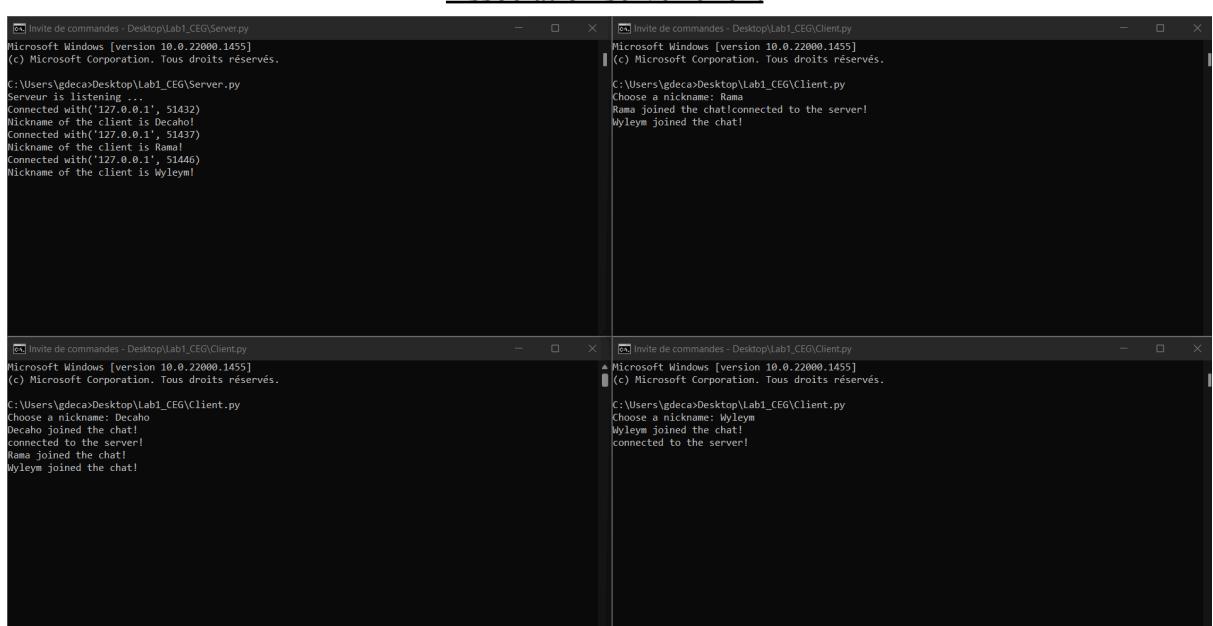
```

[...] Invite de commandes - Desktop\Lab1_CEG\Client.py
Microsoft Windows [version 10.0.22000.1455]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\gdeca>Desktop\Lab1_CEG\Client.py
Choose a nickname: Decaho
Decaho joined the chat!
connected to the server!

```

Association server-client



The image displays four terminal windows arranged in a 2x2 grid, illustrating the interaction between a server and multiple clients.

- Top Left Window:** Shows the server log. It starts with the standard Windows command prompt, followed by the server's response to the client's connection attempt. The client's nickname, "Decaho", is displayed.
- Top Right Window:** Shows the client log. It starts with the standard Windows command prompt, followed by the client's choice of nickname, "Decaho", and its confirmation that it has joined the chat and connected to the server.
- Bottom Left Window:** Shows the server log again, this time receiving connections from two additional clients: "Rama" and "Wyleym". Each client's nickname and its confirmation of joining the chat are listed.
- Bottom Right Window:** Shows the client log for the second client, "Wyleym". It shows the client choosing the nickname "Wyleym", confirming its joining of the chat, and confirming its connection to the server.

Visualisation de l'interaction entre le serveur et les différents clients

```

[invite de commandes - Desktop\Lab1_CEG\Server.py]
Microsoft Windows [version 10.0.22000.1455]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\gdeca>Desktop\Lab1_CEG\Server.py
Serveur is listening ...
Connected with '127.0.0.1', 51432
Nickname of the client is Decaho!
Connected with('127.0.0.1', 51437)
Nickname of the client is Rama!
Connected with('127.0.0.1', 51446)
Nickname of the client is Wyleym!

```



```

[Sélection invite de commandes]
Microsoft Windows [version 10.0.22000.1455]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\gdeca>

```



```

[invite de commandes - Desktop\Lab1_CEG\Client.py]
Microsoft Windows [version 10.0.22000.1455]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\gdeca>Desktop\Lab1_CEG\Client.py
Choose a nickname: Decaho
Decaho joined the chat!
connected to the server!
Rama joined the chat!
Wyleym joined the chat!
Rama left the chat !

```



```

[invite de commandes - Desktop\Lab1_CEG\Client.py]
Microsoft Windows [version 10.0.22000.1455]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\gdeca>Desktop\Lab1_CEG\Client.py
Choose a nickname: Wyleym
Wyleym joined the chat!
connected to the server!
Rama left the chat !

```

Un client quitte le server

5- Discussion

Dans ce laboratoire on a utilisé toutes les données misent à notre disposition pour mener à bien le travail demandé. Néanmoins on a eu quelques difficultés notamment la connexion entre deux clients et plus ou la connexion et la stabilisation entre client et serveur pour le résoudre nous avons utilisé les Thread pour régler ce problème. On a aussi eu un problème quand au moyen de faire sortir un client du serveur sans que le serveur ne s'interrompt, pour le régler nous pouvons juste fermer l'invité de commande et le client se fera automatiquement sortir du serveur.

6- Conclusion

Pour le laboratoire 1 l'objectif était la conception et la mise en œuvre d'une application clients et serveur de bavardage orientée connexion (TCP/IP). L'application a été conçue avec le langage de programmation Python .

En somme, ce laboratoire a été gratifiant pour nous car il nous a permis d'approfondir nos connaissances en matière de sockets, de clavardage entre clients et la connexion client-serveur. On a pu affronter et régler les différents problèmes rencontrés lors de la conception de l'application et nous nous en servirons pour les prochaines expériences.