

Programmation d'une Application de messagerie instantanée en utilisant les Sockets



Table des matières

Cahier des charges.....	2
Réalisation	3
Client :.....	3
Serveur :	4
Jeu de tests.....	5
Problèmes rencontrés	8

Cahier des charges

Notre application développée en langage C est une messagerie instantanée. Les clients s'envoient des messages qui passent d'abord par un serveur. Les commandes que peuvent envoyer le client sont:

- *liste* : pour voir la liste des connectés
- *pseudonyme <message-à-transmettre>* : pour envoyer un message au client avec le pseudonyme indiqué
- *quit* : pour quitter le tchat.

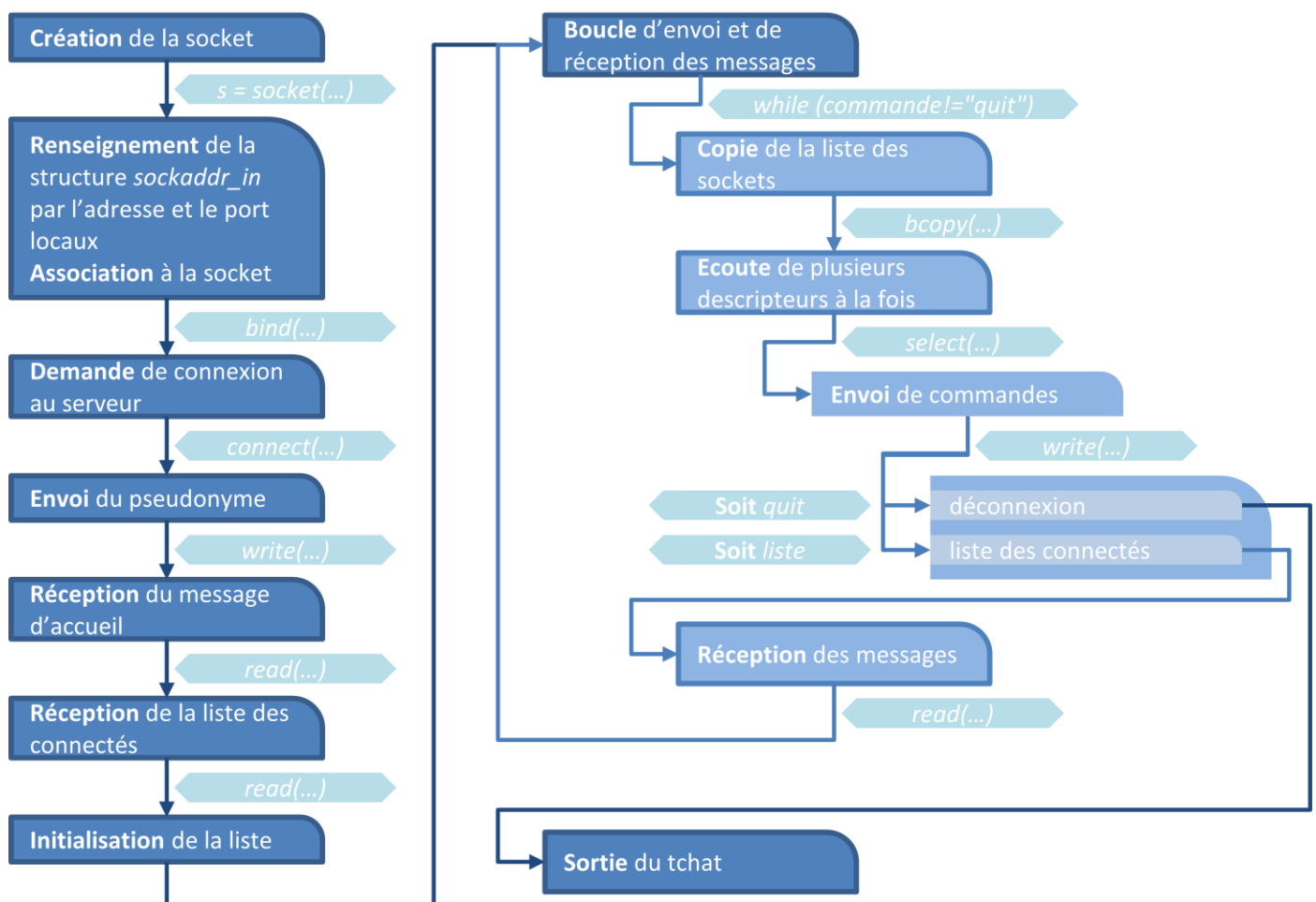
Cependant, nous avons deux contraintes :

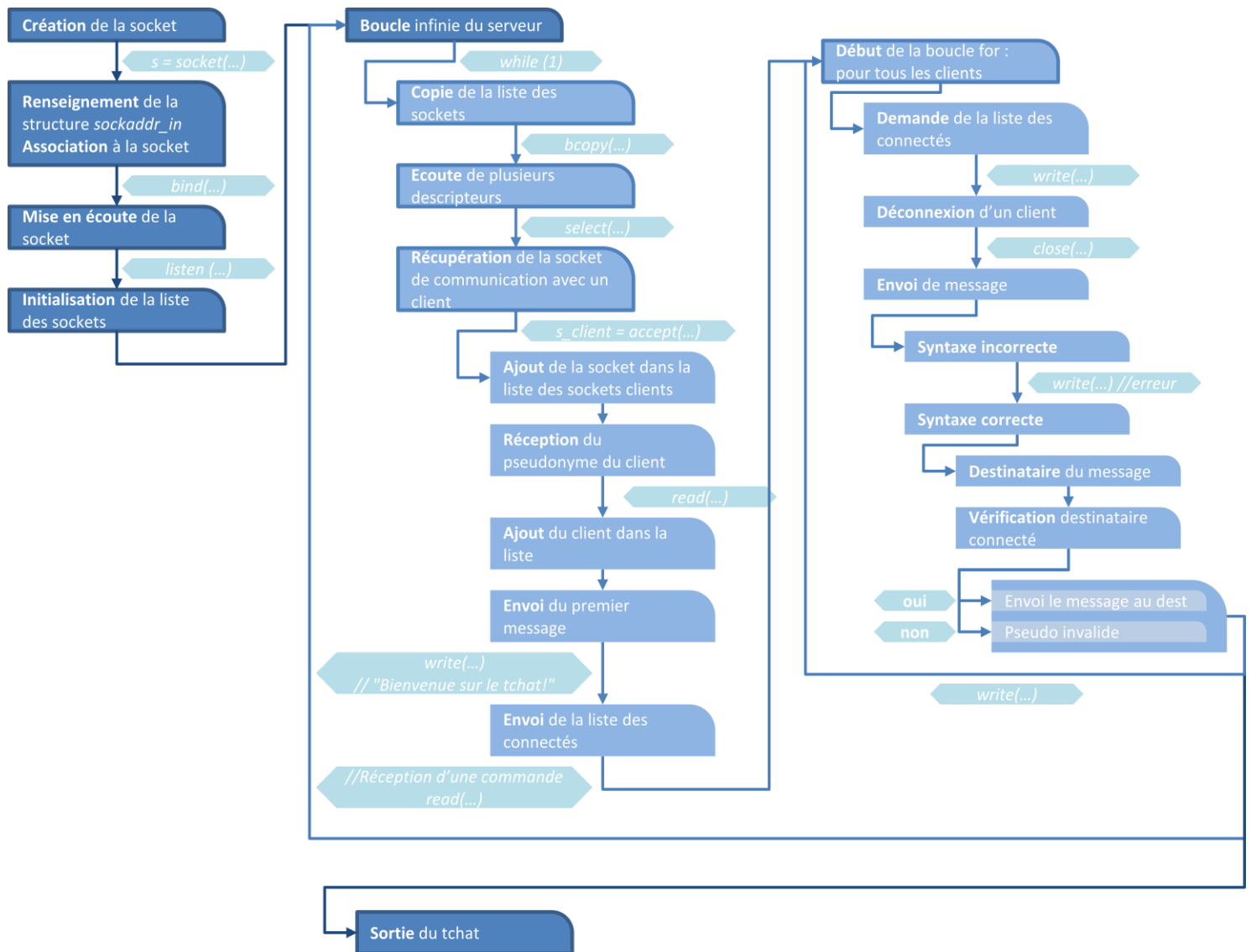
- Le protocole utilisé est le protocole TCP
- Le serveur doit rester à l'écoute des différents clients.

Réalisation

Les automates suivants résument le fonctionnement du système côté client et côté serveur.

Client :



Serveur :

Le code a été commenté pour faciliter votre compréhension

Jeu de tests

-Plusieurs clients peuvent se connectés en même temps

The image displays four terminal windows arranged in a 2x2 grid, illustrating a chat server and three clients connecting at the same time.

- Top-Left (Server):** The server terminal shows the initialization process: "Numero de port d'ecoute = 9999", "Socket creee correctement", "bind bien passe", "listen bien passe : j'ecoute...", "initialisation terminee", and "Bienvenue sur le Serveur". It then logs incoming connections: "J'ai reçu une demande du client : 127.0.0.1", "Connexion du client : yass", "J'ai reçu une demande du client : 127.0.0.1", "Connexion du client : merry", and "J'ai reçu une demande du client : 127.0.0.1", "Connexion du client : toto".
- Top-Right (Client Yass):** The client terminal shows the connection process: "yass@yass-laptop: ~/Documents/RICM4/TR/essai", "yass@yass-laptop:~/Documents/RICM4/TR/essai\$./client 127.0.0.1 9999 yass", "Socket creee correctement", "Tente une connexion au serveur...", "Connexion reussie", "Bienvenue sur le tchat!". It also displays the help menu: "*Liste : permet de voir la liste des connectes", "*quit : permet de quitter le chat", and "*Pour envoyer un message : pseudo <<message>>".
- Bottom-Left (Client Merry):** The client terminal shows the connection process: "yass@yass-laptop:~/Documents/RICM4/TR/essai\$./client 127.0.0.1 9999 merry", "Socket creee correctement", "Tente une connexion au serveur...", "Connexion reussie", "Bienvenue sur le tchat!". It also displays the help menu.
- Bottom-Right (Client Toto):** The client terminal shows the connection process: "yass@yass-laptop:~/Documents/RICM4/TR/essai\$./client 127.0.0.1 9999 toto", "Socket creee correctement", "Tente une connexion au serveur...", "Connexion reussie", "Bienvenue sur le tchat!". It also displays the help menu.

-La commande **Liste** permet de voir le liste des connectés

The image displays four terminal windows arranged in a 2x2 grid, illustrating the use of the 'Liste' command to view connected users.

- Top-Left (Server):** The server terminal shows the initialization process and incoming connections. It then displays the 'Tchat : Liste' command being entered three times.
- Top-Right (Client Yass):** The client terminal shows the connection process and the help menu. It then displays the 'Liste' command being entered, resulting in the output: "yass,merry,toto connectes".
- Bottom-Left (Client Merry):** The client terminal shows the connection process and the help menu. It then displays the 'Liste' command being entered, resulting in the output: "yass,merry,toto, connectes".
- Bottom-Right (Client Toto):** The client terminal shows the connection process and the help menu. It then displays the 'Liste' command being entered, resulting in the output: "yass,merry,toto, connectes".

-La commande **quit** permet de sortir du chat
(toto a quitté le tchat)

The image displays four terminal windows illustrating the chat application's operation:

- Top-Left (Server):** Shows the server's log of connections. It receives requests from client 127.0.0.1 (yass, merry, and toto) and successfully connects them. It also shows the 'quit' command being sent by toto, resulting in 'toto est deconnecte'.
- Top-Right (Client Yass):** Shows the client's perspective. It successfully connects to the server and receives the chat list: 'yass,merry,toto connectes'.
- Bottom-Left (Client Merry):** Shows the client's perspective. It successfully connects to the server and receives the chat list: 'yass,merry,toto, connectes'.
- Bottom-Right (Client Toto):** Shows the client's perspective. It successfully connects to the server and receives the chat list. It then enters the 'quit' command, which is shown in the input line.

Les clients peuvent s'envoyer des messages à condition de respecter la syntaxe spécifiée :

-Echange entre toto et yass seulement

The image displays four terminal windows illustrating the chat application's operation:

- Top-Left (Server):** Shows the server's log of connections. It receives requests from client 127.0.0.1 (merry and toto) and successfully connects them. It also shows the 'quit' command being sent by toto, resulting in 'toto est deconnecte'.
- Top-Right (Client Yass):** Shows the client's perspective. It successfully connects to the server and receives the chat list: 'yass,merry,toto connectes'.
- Bottom-Left (Client Merry):** Shows the client's perspective. It successfully connects to the server and receives the chat list: 'yass,merry,toto, connectes'.
- Bottom-Right (Client Toto):** Shows the client's perspective. It successfully connects to the server and receives the chat list. It then enters the 'quit' command, which is shown in the input line.

-Les trois clients peuvent communiquer (yass, toto et titi)

The image shows four terminal windows arranged in a 2x2 grid, illustrating a chat application. Each window has a title bar with icons and the text 'yass@yass-laptop: ~/Documents/RICM4/TR/essai'. The top-left window is labeled 'Serveur' and shows the server's log of connections and messages. The top-right window is labeled 'Client Yass' and shows the client's connection process and received messages. The bottom-left window is labeled 'Client Toto' and shows the client's connection process and received messages. The bottom-right window is labeled 'Client Titi' and shows the client's connection process and received messages. All clients have successfully connected to the server and can send and receive messages.

```

yass@yass-laptop: ~/Documents/RICM4/TR/essai
Fichier Édition Affichage Terminal Aide
J'ai reçu une demande du client : 127.0.0.1
Connexion du client : yass

Tchat : quit
marie est deconnecte
J'ai reçu une demande du client : 127.0.0.1
Connexion du client : toto

J'ai reçu une demande du client : 127.0.0.1
Connexion du client : titi

Tchat : toto comment va
Tchat : titi comment va
Tchat : titi salut
Tchat : yass salut
Tchat : toto coucou
Tchat : yass coucou

yass@yass-laptop: ~/Documents/RICM4/TR/essai$ ./client 127.0.0.1 9998 yass
Socket creee correctement
Tente une connexion au serveur...
Connexion reussie
Bienvenue sur le tchat!

*Liste : permet de voir la liste des connectes
*quit : permet de quitter le chat
*Pour envoyer un message : pseudo <<message>>

toto comment va
titi comment va
toto : salut10

titi : coucou

yass@yass-laptop: ~/Documents/RICM4/TR/essai$ ./client 127.0.0.1 9998 toto
Socket creee correctement
Tente une connexion au serveur...
Connexion reussie
Bienvenue sur le tchat!

*Liste : permet de voir la liste des connectes
*quit : permet de quitter le chat
*Pour envoyer un message : pseudo <<message>>

yass : comment va

titi salut
yass salut
titi : coucou

yass@yass-laptop: ~/Documents/RICM4/TR/essai$ ./client 127.0.0.1 9998 titi
Socket creee correctement
Tente une connexion au serveur...
Connexion reussie
Bienvenue sur le tchat!

*Liste : permet de voir la liste des connectes
*quit : permet de quitter le chat
*Pour envoyer un message : pseudo <<message>>

yass : comment va

toto : salut10

toto coucou
yass coucou
  
```

-Entrée d'une commande invalide

The image shows four terminal windows arranged in a 2x2 grid, illustrating a chat application where clients enter invalid commands. Each window has a title bar with icons and the text 'yass@yass-laptop: ~/Documents/RICM4/TR/essai'. The top-left window is labeled 'Serveur' and shows the server's log of connections and messages, including 'commande invalide'. The top-right window is labeled 'Client Yass' and shows the client's connection process and received messages, including 'La commande que vous avez entree n est pas valide'. The bottom-left window is labeled 'Client Merry' and shows the client's connection process and received messages, including 'La commande que vous avez entree n est pas valide'. The bottom-right window is labeled 'Client Toto' and shows the client's connection process and received messages, including 'La commande que vous avez entree n est pas valide'. All clients have successfully connected to the server but are unable to send certain commands.

```

yass@yass-laptop: ~/Documents/RICM4/TR/essai
Fichier Édition Affichage Terminal Aide
Tchat : Liste
Tchat : yass slt
Tchat : quit
yass est deconnecte
J'ai reçu une demande du client : 127.0.0.1
Connexion du client : yass

Tchat : salut
commande invalide

Tchat : coucou
commande invalide

Tchat : hello
commande invalide

yass@yass-laptop: ~/Documents/RICM4/TR/essai$ ./client 127.0.0.1 9999 merry
Socket creee correctement
Tente une connexion au serveur...
Connexion reussie
Bienvenue sur le tchat!

*Liste : permet de voir la liste des connectes
*quit : permet de quitter le chat
*Pour envoyer un message : pseudo <<message>>

liste
yass,merry,toto, connectes
liste
merry,toto,yass, connectes
yass slt
hello
La commande que vous avez entree n est pas valide

yass@yass-laptop: ~/Documents/RICM4/TR/essai$ ./client 127.0.0.1 9999 yass
Socket creee correctement
Tente une connexion au serveur...
Connexion reussie
Bienvenue sur le tchat!

*Liste : permet de voir la liste des connectes
*quit : permet de quitter le chat
*Pour envoyer un message : pseudo <<message>>

salut
La commande que vous avez entree n est pas valide

yass@yass-laptop: ~/Documents/RICM4/TR/essai$ ./client 127.0.0.1 9999 toto
Socket creee correctement
Tente une connexion au serveur...
Connexion reussie
Bienvenue sur le tchat!

*Liste : permet de voir la liste des connectes
*quit : permet de quitter le chat
*Pour envoyer un message : pseudo <<message>>

coucou
La commande que vous avez entree n est pas valide
  
```


-Déconnexion des 3 clients et le serveur est en attente d'une reconnexion

```

Connexion du client : toto
J'ai reçu une demande du client : 127.0.0.1
Connexion du client : titi

Tchat : toto comment va
Tchat : titi comment va
Tchat : titi salut
Tchat : yass salut
Tchat : toto coucou
Tchat : yass coucou
Tchat : quit
yass est deconnecte
Tchat : quit
titi est deconnecte
Tchat : quit
toto est deconnecte

yass@yass-laptop:~/Documents/RICM4/TR/essai$ ./client 127.0.0.1 9998 yass
Socket creee correctement
Tente une connexion au serveur...
Connexion reussie
Bienvenue sur le tchat!

*Liste : permet de voir la liste des connectes
*quit : permet de quitter le chat
*Pour envoyer un message : pseudo <<message>>

toto comment va
titi comment va
toto : salut!
titi : coucou
quit
yass@yass-laptop:~/Documents/RICM4/TR/essai$

yass@yass-laptop:~/Documents/RICM4/TR/essai$ ./client 127.0.0.1 9998 toto
Socket creee correctement
Tente une connexion au serveur...
Connexion reussie
Bienvenue sur le tchat!

*Liste : permet de voir la liste des connectes
*quit : permet de quitter le chat
*Pour envoyer un message : pseudo <<message>>

yass : comment va

titi salut
yass salut
titi : coucou

quit
yass@yass-laptop:~/Documents/RICM4/TR/essai$

yass@yass-laptop:~/Documents/RICM4/TR/essai$ ./client 127.0.0.1 9998 yass
Socket creee correctement
Tente une connexion au serveur...
Connexion reussie
Bienvenue sur le tchat!

*Liste : permet de voir la liste des connectes
*quit : permet de quitter le chat
*Pour envoyer un message : pseudo <<message>>

yass : comment va

toto : salut!
toto coucou
yass coucou
quit
yass@yass-laptop:~/Documents/RICM4/TR/essai$

```

Problèmes rencontrés

La troisième approche recommandée est un peu difficile à mettre en œuvre. En effet, on a un seul processus qui surveille l'ensemble des sockets et effectue les actions nécessaires. Ce qui fait qu'on a un automate assez grand et il faut intégrer le maximum des cas pour traiter les erreurs.

Les restrictions par rapport au cahier des charges sont qu'on n'affiche pas la liste des connectés à la connexion des clients. Le client doit écrire la commande Liste pour avoir ces informations. Et aussi si un client se déconnecte, on n'informe pas ses contacts.