

Crash Tests

Nous avons répertorié toutes les actions que l'on pouvait faire sur notre serveur chat et nous les avons classées dans la liste ci-dessous pour voir si tout fonctionnait correctement. Nous avons également ajouté des captures Wireshark pour bien montrer les envois de paquets à partir de notre chat serveur.

Login function (command line):

Ne fonctionne pas temps que l'on ne tape pas 1 ou 2

1) Se connecter

- Se connecter avec un bon user bon pwd
- Se connecter avec un bon user mauvais pwd, affiche « wrong credentials » et nous redemande 1 ou 2
- Se connecter avec un mauvais user bon pwd, affiche « wrong credentials » et nous redemande 1 ou 2
- Se connecter avec un mauvais user mauvais pwd, affiche « wrong credentials » et nous redemande 1 ou 2
- Voir si on peut bien sortir de la boucle se connecter, si on n'est pas connecté ça nous demande 1 ou 2 donc c'est bon
- Se connecter avec un client déjà connecté, dès que l'on tape son username, il nous dit déjà connecté avec un message, on revient sur 1 ou 2

2) Créer un user

- Créer un user avec un user qui existe pas et un pwd, ça créer le user et on est direct dans le chat public, si on fait listU derrière le client est bien dedans , et aussi dans la db
- Créer un user avec un user qui existe et un pwd différent, message qui nous dit que le username existe déjà, donc renvoie sur 1 ou 2 (juste pas de retour à la ligne)
- Créer un user avec un user qui existe et un pwd qui existe, message qui nous dit que le username existe déjà, donc renvoie sur 1 ou 2 (juste pas de retour à la ligne)
- Verifier une fois créer qu'on est bien dans le chat

Server function (command line):

1) #Help (list command)

- Help fonctionne correctement
- Help + quelque chose ne fonctionne pas
- Help minuscule ne fonctionne pas

2) #Exit (server shutdown)

- Exit fonctionne

- Exit + quelque chose ne fonctionne pas
- Exit minuscule fonctionne pas
- Exit pendant qu'on crée un user fonctionne
- Exit pendant que des gens sont en chat privé fonctionne (room et privé)
- Exit pendant qu'on écrit un message fonctionne

3) #Kill <user>

- Kill avec un bon username fonctionne
- Kill quand il est dans une room (ou chat privé) fonctionne
- Kill quand c'est un mauvais username fonctionne pas

4) #ListU (list of users in a server)

- ListU affiche bien la liste des clients connecté (même ceux en chat privé, même ceux dans une room)
- Connecte un client, puis le déconnecte, on vérifie qu'il est plus dans listU

5) #ListF (list of files in a server)

- ListF fonctionne normalement
- Ne fonctionne pas si mal écrit (minuscule, avec un mot après)

6) # Private <user> (private chat with another user)

- Private fonctionne normalement :
 - Créer bien un chat privé
 - Reçois toujours les messages publics
 - Envoie seulement au client en privé
- Ne peut pas créer de chat privé avec toi-même
- Ne peut pas créer de chat privé avec un client pas connecté, ou existe pas

7) #Alert <all users>

- Alerte fonctionne bien
- Alerte ne fonctionne pas quand minuscule ou mal écrit
- Fonctionne avec des messages différents

Client function (command line):

VERIFIER QUE TOUTES LES COMMANDES FONCTIONNE A N'IMPORTE QUEL MOMENT

1) #Help (list command)

- Help fonctionne correctement
- Help fonctionne pas si mal écrit (minuscule, espace, mot après)
- Help fonctionne tout le temps

2) #Exit (client exit)

- Exit fonctionne bien
- Ne fonctionne pas si mal écrit
- Fonctionne quand on est en chat privé ou room
- Informe bien tout le monde (serveur aussi) que je suis partie

3) #ListU (list of users in a server)

- ListU fonctionne bien
- Ne fonctionne pas si mal écrit
- Afficher plus un user s'il s'était connecté puis déconnecté
- Fonctionne meme dans un chat ou une room

4) #ListF (list of files in a server)

- ListF fonctionne bien
- Ne fonctionne pas si mal écrit
- Fonctionne meme dans un chat ou dans une room

5) #TrfU (Upload file transfer to a server)

- Fonctionne avec un fichier dans le répertoire courant
- Fonctionne avec un fichier dans un autre répertoire (format C:\Users\...) espaces OK
- Renvoi "Error with your file" si le fichier n'existe pas

6) #TrfD (transfer Download file to a server)

- Télécharge le fichier demandé
- Renvoie "Error with your file" si le fichier n'existe pas

7) Private <user> (private chat with another user)

- Private fonctionne normalement :
 - Crée bien un chat privé
 - Reçoit toujours les messages publics on est notifié que c'est public ou privé
 - Envoie seulement au client en privé
 - Si un client quitte, l'autre est notifié
 - Si un client quitte il n'envoie plus de message privé
 - Si un client A quitte mais pas le B, B peut toujours envoyer en privé à A
 - Si client A quitte le serveur, B est encore dans un chat privé donc quand il écrit personne ne reçoit
- Peut créer un chat privé avec toi-même, on peut et du coup quand on parle on reçoit ses propres messages, on reçoit quand meme les messages publics
- Ne peut pas créer de chat privé avec un client pas connecté, ou inexistant
- Si un client est en privé, se déconnecte et reconnecte, le client qui se reconnecte est en public, mais l'autre est toujours en privé avec le client qui se reconnecte
- Checker quand 3 clients veulent faire un chat privé ce que ca fait :

- Quand un client A qui est en chat privé avec B en créer un autre avec C, alors A parle en privé à C, C parle en privé à A, et B à A (mais du coup pas A à B)
 - Quand le client qui a 2 chats privés fait public, il revient direct en public pas dans l'ancien chat privé
 - Quand un client qui a un chat privé fait public, ça marche
- Quand le client A est déjà en private mais veut faire un autre private, cela crée un nouveau private avec le client C, l'ancien privé B peut écrire en privé et A reçoit les messages de B mais A ne peut pas écrire en privé à B vu qu'il écrit au dernier client avec qu'il s'est mis en privé, à savoir C.

8) Public (back to the public)

- Public fonctionne bien
- Ne fonctionne pas si mal tapé
- Fonctionne meme avec plusieurs chats privés
- Notifie l'autre quand on passe en public
- Public si pas dans un chat privé ou room, cela ne change rien
- Si on exit le serveur alors qu'on était en private, ça prévient bien l'autre client qu'on a exit le serveur mais pas qu'on est encore en privé avec lui

9) #Ring <user> (notification if the user is logged in)

- Fonctionne correctement
- Ring un user meme dans un chat ou en private, même en privé on peut ring un client qui est en public, et inversement
- Ring pas si user n'existe pas ou mal tapé
- Ring soi-meme, on peut se ring sois même

Room function (command line):

1) #CreateRoom<room_name> <user1> <user2> ...

- Fonctionne correctement
- Ne fonctionne pas si mal tapé
- Ne fonctionne pas si nom de la room existe déjà
- Si le nom de la room contient des espaces, le premier mot sera considéré comme le nom de la room et les autres seront considéré comme des noms de user
- Ne fonctionne pas si on essaye de créer une room avec un seul user (autre que nous)
- Ne fonctionne pas si on essaye de créer une room avec uniquement 1 user mais en mettant deux fois ou plus son nom (#CreateRoom elite user1 user1)
- Ne fonctionne pas si le room_name = username d'un user (connecté ou pas)
- Ne fonctionne pas si on écrit son propre nom dans la liste des users

2) #KickRoom <room_name> <user>

- Fonctionne correctement
- Ne fonctionne pas si mal tapé

- Ne fonctionne pas si user qu'on veut kick n'est pas dans la room ou n'est pas connecté
- Ne fonctionne pas si user n'est pas admin de la room
- Ne fonctionne pas si user n'appartient pas à la room
- Fonctionne pas si la room n'existe pas
- User que l'on kick doit être redirigé vers chat public
- Si la room se supprime parce qu'il reste que trop peu de user, tout le monde doit être redirigé vers le chat public

3) #LeaveRoom <room_name>

- Fonctionne correctement
- Ne fonctionne pas si mal tapé
- Ne fonctionne pas si on n'appartient pas à la room
- Ne fonctionne pas si la room n'existe pas
- On doit être redirigé vers le chat public après avoir leave
- On ne doit pas pouvoir join la room après l'avoir leave

4) #JoinRoom <room_name>

- Fonctionne correctement
- Ne fonctionne pas si mal tapé
- Ne fonctionne pas si on n'appartient pas à la room
- Ne fonctionne pas si la room n'existe pas

5) #ListClientRoom <room_name>

- Fonctionne correctement
- Ne fonctionne pas si mal tapé
- Ne fonctionne pas si user n'appartient pas à la room
- Ne fonctionne pas si la room n'existe pas

6) #ListRoom

- Fonctionne correctement
- Ne fonctionne pas si mal tapé
- Ne fonctionne pas si user n'appartient pas à la room
- Ne fonctionne pas si la room n'existe pas

7) #AddRoom <room_name> <user>

- Fonctionne correctement
- Ne fonctionne pas si mal tapé
- Ne fonctionne pas si user qu'on veut add est dans la room ou n'est pas connecté
- Ne fonctionne pas si user n'est pas admin de la room
- Ne fonctionne pas si la room n'existe pas

Autres :

- Envoyer un message de plus de 280 caractères, message nous indiquant que le message est trop long, personne ne reçoit le message
- Transférer les fichiers client to client et client to server
- Database est bien sur le serveur
- Encryption fonctionne bien

Quand on envoie un message :

The image displays two side-by-side Wireshark network capture screenshots. Both captures are filtered for 'loob' (localhost) traffic. The left capture shows a sequence of four packets: a SYN (Seq: 64733), an ACK (Seq: 64733, Ack: 64733), a SYN (Seq: 64733), and an ACK (Seq: 64733, Ack: 64733). The right capture shows a sequence of four packets: a SYN (Seq: 64733), an ACK (Seq: 64733, Ack: 64733), a SYN (Seq: 64733), and an ACK (Seq: 64733, Ack: 64733). Both captures show the data field of the packets, which contains the message 'Hello World'.