# Présentation du projet de programmation réseau :

# Messagerie instantanée

# Sommaire

I - Protocole

II – L’implémentation

1. Fonctions implémentées
2. Fonctions non-implémentées par faute de temps

III – Les limitations & bugs

IV – Manuel utilisateur

# I – Protocole

* Header :
  + Commande : 1 octet
  + IdUser : 4 octets
  + Timestamp : 4 octets
  + IdSalon : 1 octet
  + Taille : 4 octets
  + NumMessage : 4 octets
* Data :
  + Chaine de caractères de taille 140

### Entête du message :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Commande | IdUser | Timestamp | IdSalon | Size | NumMsg | Data |

### Commandes clientes:

* 0 : Connect

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | Time | 0 | S | 0 | Username |

Le message Connect renvoie 1 si la connexion est autorisée et 0 si l’utilisateur est déjà connecté ou que le serveur refuse la connexion.

* 1 : Join

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | IdUser | Time | 0 | S | 0 | Roomname |

Le message Join ne fonctionne que si l’utilisateur est déjà connecté. Le serveur répondra 0 si l’utilisateur n’est pas connecté où s’il est déjà présent dans ce salon. Sinon, il retourne l’IdSalon.

* 2 : Say

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2 | IdUser | Time | IdSalon | S | 0 | Message |

Le message Say retourne 0 si l’utilisateur n’est pas connecté ou n’est pas dans un salon. Il retourne 1 sinon.

* 3 : Leave

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | IdUser | Time | IdSalon | S | 0 | 0 |

Le message Leave renvoie 0 si l’utilisateur n’est pas connecté ou n’est pas dans un salon. Il renvoie 1 sinon.

* 4 : Disconnect

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4 | IdUser | Time | 0 | S | 0 | 0 |

Le message Disconnect renvoie 0 si l’utilisateur n’est pas connecté et 1 sinon.

* 5 : Ack

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 5 | IdUser | Time | IdSalon | S | 0 | 1 |

Renvoie 0 si le message n’a pas été acquitté. 1 sinon.

* 6 : Alive

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 6 | IdUser | Time | 0 | S | 0 | 0 |

Le message Alive renvoie tout le temps 1.

### Commandes serveur :

* 5 : Ack

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 5 | IdUser | Time | 0 | S | 0 | 1 |

Renvoie 1 si OK

* 7 : MessageSrv

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 7 | IdUser | Time | IdSalon | S | NumMsg | Message |

# II – L’implémentation

## Fonctions implémentées

Connexion multi-utilisateurs

Présence d’une fonction de vérification de présence : Toutes les 15 secondes, un message est envoyé par le client au serveur afin de lui dire qu’il est bien vivant.

## Fonctions non implémentées par faute de temps

Filtre anti-spam

Filtre anti-grossièreté

# Les limitations & bugs

Taille des messages limitée à 140

# Manuel utilisateur

Dans un premier temps, il faut extraire les fichiers qui sont situés dans l’archive .tar.gz. Une fois que les fichiers sont extrais, il faut ensuite créer les exécutables :

make

Deux exécutables (Client et Serveur) sont alors créés.

Il faut ensuite lancer dans un terminal le serveur seul ou le lancer en tâche de fond

./Serveur || ./Serveur &

Et le client

./Client 127.0.0.1

Une fois que les deux exécutables sont lancés, les tests peuvent commencer.

Plusieurs choses sont à tester. :

* Connexion à une room : JOIN nom\_de\_la\_room
* Parler dans une room : SAY texte || texte (si la chaîne de caractères ne commence pas par une commande, la commande SAY est mise par défaut !).