Protocole IRCChat

**But du projet :** Créer un chat en langage C avec la possibilité de rejoindre différents salons sous un pseudonyme choisi par l’utilisateur.

# Client

Un client est une interface de connexion au serveur de chat.

Chaque client peut rejoindre un ou plusieurs salons et discuter avec les personnes présentes sur ceux-ci.

Le client peut interagir avec le serveur par le biais des commandes définies dans le protocole.

Le client doit signaler son activité de façon régulière au serveur pour éviter d’être considéré comme déconnecté

# Serveur

Le serveur est en dialogue permanent avec les clients.

Il interprète les commandes du protocole et redistribue les messages sur les différents salons.

# Architecture réseau

# Commandes du protocole

Un message commençant par un / est considéré comme une commande au serveur.

Sinon il est considéré comme un message à poster sur le salon courant.

## Etablissement de la connexion

SERVER <Adresse > <Port> : Connexion au serveur

## Opération sur les canaux

NICK <Pseudonyme> : Modification de son pseudonyme

JOIN <Salon> : rejoindre un salon ou le créer si il n’existe pas

PART : Quitter le salon courant

QUIT : Se déconnecter du serveur

LIST : liste tous les salons ouverts

HELP : Affiche la liste des commandes possibles

# Détails des commandes

## Commande MESSAGE

Commande : Aucune

Paramètres : Aucun paramètre.

La commande MESSAGE est utilisée lorsqu’un client souhaite transmettre un message à un salon.

Réponse possible du serveur :

* ACK\_MESSAGE

Trame client -> server

|  |  |  |
| --- | --- | --- |
| MESSAGE | <Contenu> | <Salon> |

Trame server->client

|  |
| --- |
| ACK\_MESSAGE |

Trame server -> autres clients

|  |  |  |
| --- | --- | --- |
| MESSAGED | <Expéditeur> | <Contenu> |

## Commande SERVER

Commande : SERVER

Paramètres : <Adresse IP> <Port> <pseudonyme>

La commande SERVER est utilisée par le client pour se connecter au server et signaler sa présence.

Réponse possible du serveur :

* ERR\_NICKALREADYUSED
* ACK\_CONNECTED

Utilisation possible :

/SERVER 127.0.0.1 1501 Aveys

Trame client -> server

|  |
| --- |
| CONNECT |

Trame server->client

|  |
| --- |
| ACK\_CONNECTED |

## Commande NICK

Commande : NICK

Paramètres : <pseudonyme>

La commande NICK permet de changer de pseudonyme pendant l’utilisation du chat.

Le pseudonyme ne doit pas contenir de caractères spéciaux ni d’espace.

Réponse possible du serveur :

* ERR\_NICKALREADYUSED
* ACK\_NICKMODIFIED

Utilisation possible :

/NICK VValot

Trame client -> server

|  |  |
| --- | --- |
| NICK | <Pseudonyme> |

Trame server->client

|  |
| --- |
| ACK\_NICKMODIFIED |

Trame server -> autres clients

|  |  |  |
| --- | --- | --- |
| NICKMODIFIED | <Ancien> | <Nouveau> |

## Commande JOIN

Commande : JOIN

Paramètres : <salon>

La commande JOIN est utilisée lorsqu’un client souhaite rejoindre un salon.

Si le salon existe, le client rejoint le salon sinon celui-ci est crée par le serveur.

Réponse possible du serveur :

* ACK\_JOIN

Utilisation possible :

/JOIN salonVoiture

Trame client -> server

|  |  |
| --- | --- |
| JOIN | <Salon> |

Trame server->client

|  |
| --- |
| ACK\_JOIN |

Trame server -> autres clients

|  |  |
| --- | --- |
| CHANJOINED | <pseudonymes> |

## Commande PART

Commande : PART

Paramètres : Aucun paramètre.

La commande PART est utilisée lorsqu’un client souhaite quitter un salon.

Réponse possible du serveur :

* ACK\_PART
* ERR\_NOCHANNELJOINED

Utilisation possible :

/PART

Trame client -> server

|  |  |
| --- | --- |
| PART | <Salon> |

Trame server->client

|  |
| --- |
| ACK\_PART |

Trame server -> autres clients

|  |  |
| --- | --- |
| CHANLEAVED | <pseudonymes> |

## Commande QUIT

Commande : QUIT

Paramètres : Aucun paramètre.

La commande QUIT est utilisée lorsqu’un client souhaite se déconnecter du serveur.

Réponse possible du serveur :

* ACK\_QUIT

Utilisation possible :

/QUIT

Trame client -> server

|  |  |
| --- | --- |
| QUIT |  |

Trame server->client

|  |
| --- |
| ACK\_QUIT |

Trame server -> autres clients

|  |  |
| --- | --- |
| CHANLEAVED | <pseudonymes> |

## Commande LIST

Commande : LIST

Paramètres : Aucun paramètre.

La commande LIST est utilisée lorsqu’un client souhaite connaître la liste des salons auxquels il est rattaché.

Réponse possible du serveur :

* ACK\_LIST

Utilisation possible :

/LIST

Trame client -> server

|  |
| --- |
| LIST |

Trame server->client

|  |  |
| --- | --- |
| ACK\_LIST | <Liste des salons> : Char [][] |

## Commande ALIVE

Commande : Aucune

Paramètres : Aucun paramètre.

La commande ALIVE est utilisée lorsqu’un client souhaite signifier au serveur qu’il est toujours connecté.

Réponse possible du serveur :

* ACK\_ALIVE

Trame client -> server

|  |
| --- |
| ALIVE |

Trame server->client

|  |
| --- |
| ACK\_ALIVE |

## Commande HELP

Commande : HELP

Paramètres : Aucun paramètre.

La commande HELP est utilisée lorsqu’un client souhaite connaître la liste des commandes possibles. Les commandes et leurs paramètres sont alors affichées.

Réponse possible du serveur :

* ACK\_HELP

Utilisation possible :

/HELP

Trame client -> server

|  |
| --- |
| HELP |

Trame server->client

|  |  |
| --- | --- |
| ACK\_HELP | <Liste des commandes> : String |

# Liste des réponses serveur

* ERR\_NICKALREADYUSED : Pseudonyme déjà utilisé
* ERR\_NOCHANNELJOINED : Aucun salon rejoint

# Liste des acquittements serveur

* ACK\_CONNECTED : Correctement connecté
* ACK\_NICKMODIFIED : Pseudonyme correctement modifié
* ACK\_JOIN : Salon rejoint
* ACK\_PART : Salon quitté
* ACK\_QUIT : Correctement déconnecté
* ACK\_LIST : Liste des salons
* ACK\_HELP : Aide aux commandes

# Structure technique

Structure Client

* Char nickname[]
* Sockaddr\_in socket\_addr

Structure Salon

* Char name[]
* Client listeClients[]

Structure Message

* String message[]
* String SalonCible