

Rapport Projet PSE

Programmation Système

Application client-serveur orientée "réseau social"

Imane MOUMOUN

Table des matières

Description du projet :.....	2
Organisation du programme	2
Mode opératoire :	3
1. Installation.....	3
2.Exécution des programmes.....	4
Déroulement du programme	4

Description du projet :

Dans le cadre de ce projet, nous développons une application client-serveur en langage C orientée "réseau social". L'objectif est de mettre en place un système de communication en temps réel entre plusieurs utilisateurs connectés au serveur.

L'application proposera les fonctionnalités suivantes :

- **Messagerie** : les utilisateurs ont la possibilité d'échanger des messages soit de manière privée, soit via des messages publics visibles par tous les utilisateurs connectés.
- **Système de vote** : les utilisateurs ont la possibilité de participer à un vote en répondant par "OUI" ou "NON", avec une seule participation autorisée par utilisateur. Les résultats sont mis à jour en temps réel, permettant à chacun de suivre l'évolution des réponses au fur et à mesure.

Ce projet vise à implémenter des concepts de programmation réseau en C notamment la gestion des connexions simultanées et l'échange de données via des sockets. L'application repose sur une architecture multi-threadée. Un thread principal du serveur assure la collecte des messages des utilisateurs et leur redistribution aux destinataires concernés. Côté clients, la réception des informations provenant du serveur est prise en charge par un thread dédié, garantissant une communication fluide et efficace.

Organisation du programme

pse.h : Nous avons choisi d'inclure le fichier **pse.h**, qui regroupe l'ensemble des bibliothèques nécessaires au bon fonctionnement de notre application serveur/client.

serveur.c : Le fichier `serveur.c` contient le programme principal qui fait fonctionner le serveur de discussion. Son rôle est d'écouter les connexions des clients, de leur permettre de s'identifier, puis d'échanger des messages entre eux. Chaque client est géré dans un thread séparé, ce qui permet à plusieurs utilisateurs d'être connectés en même temps. Le serveur comprend aussi un système de vote où un utilisateur peut poser une question, et les autres peuvent voter pour ou contre.

client.c : Le fichier `client.c` correspond à l'application que chaque utilisateur utilise pour se connecter au serveur. Lors du lancement, l'utilisateur indique l'adresse du serveur, le port à utiliser, et un nom d'utilisateur. Une fois connecté, il peut envoyer des messages publics ou privés, participer à des votes, ou simplement voir qui est connecté.

journal.log : Le fichier `journal.log` permet d'enregistrer les connexions et déconnexions des utilisateurs, l'envoi de messages publics et privés, ainsi que le déroulement des votes, depuis leur lancement jusqu'à la participation des utilisateurs et l'annonce des résultats une fois le vote terminé. Ce fichier est généré automatiquement par le serveur s'il n'existe pas. Par ailleurs, à chaque redémarrage du serveur, son contenu est effacé afin de garantir un historique propre et à jour.

Mode opératoire :

1. Installation

1. Téléchargez l'archive du projet.
2. Ouvrez un terminal et placez-vous dans votre répertoire personnel.
3. Extrayez l'archive à l'aide de la commande suivante :

```
> tar xvf MOUMOUN.IGOUZOULENE.tar
```

4. Placez-vous dans le répertoire du projet

> `cd MOUMOUN.IGOUZOULENE/projet`

5. Exécutez le Makefile et créez les exécutables via la commande:

> `make`

A ce stade, deux exécutables ont été créés : serveur & client.

2.Exécution des programmes

Pour démarrer un serveur vous pouvez exécuter :

> `./serveur 2000`

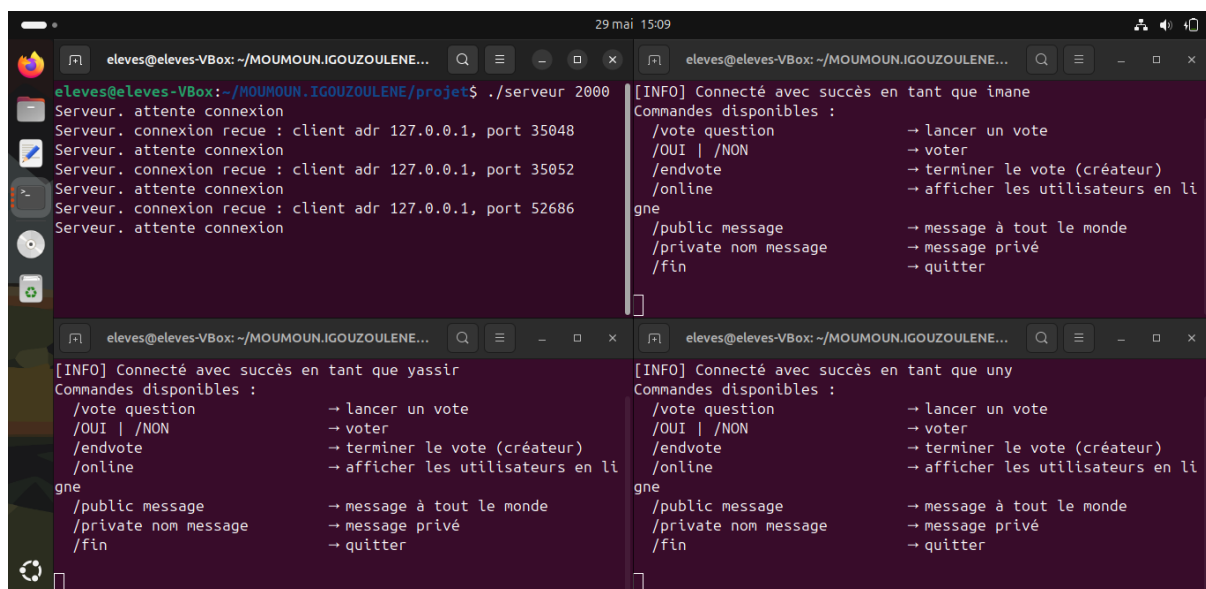
Remarque : le numéro du port peut être modifié.

Ensuite, Les clients peuvent se connecter sur des instances de terminal différentes

> `./client localhost 2000 identifiant`

Déroulement du programme

Lors de la connexion des utilisateurs, le message suivant s'affiche avec la liste des commandes disponibles:



The screenshot displays three terminal windows from a Linux desktop environment. The top window shows the server being started with the command `./serveur 2000`. It logs several connection attempts from clients at IP 127.0.0.1 on ports 35048, 35052, and 52686. The bottom two windows show clients connecting to the server. Each client window displays a success message and a list of available commands with their functions: `/vote question` (lancer un vote), `/OUI | /NON` (voter), `/endvote` (terminer le vote (créateur)), `/online` (afficher les utilisateurs en ligne), `/public message` (message à tout le monde), `/private nom message` (message privé), and `/fin` (quitter).

```
eleves@eleves-VBox: ~/MOUMOUN.IGOUZOULENE/projet$ ./serveur 2000
[INFO] Connecté avec succès en tant que imane
Commandes disponibles :
/vote question      → lancer un vote
/OUI | /NON         → voter
/endvote            → terminer le vote (créateur)
/online             → afficher les utilisateurs en li
gne
/public message     → message à tout le monde
/private nom message → message privé
/fin                → quitter

eleves@eleves-VBox: ~/MOUMOUN.IGOUZOULENE...$
[INFO] Connecté avec succès en tant que yassir
Commandes disponibles :
/vote question      → lancer un vote
/OUI | /NON         → voter
/endvote            → terminer le vote (créateur)
/online             → afficher les utilisateurs en li
gne
/public message     → message à tout le monde
/private nom message → message privé
/fin                → quitter

eleves@eleves-VBox: ~/MOUMOUN.IGOUZOULENE...$
[INFO] Connecté avec succès en tant que uny
Commandes disponibles :
/vote question      → lancer un vote
/OUI | /NON         → voter
/endvote            → terminer le vote (créateur)
/online             → afficher les utilisateurs en li
gne
/public message     → message à tout le monde
/private nom message → message privé
/fin                → quitter
```

A l'aide de la commande `/online`, l'utilisateur peut voir la liste des personnes connectés :

```

eleves@eleves-VBox: ~/MOUMOUN.IGOUZOULENE...
eleves@eleves-VBox:~/MOUMOUN.IGOUZOULENE/projet$ ./serveur 2000
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 35048
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 35052
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 52686
Serveur. attente connexion

/online → afficher les utilisateurs en li
/public message → message à tout le monde
/private nom message → message privé
/fin → quitter

/online
[ONLINE] Liste utilisateurs en ligne:
1) imane
2) yassir
3) uny
  
```

A l'aide de la commande `/public message`, l'utilisateur* peut envoyer un message visible à tous les autres personnes connectées* :

```

eleves@eleves-VBox: ~/MOUMOUN.IGOUZOULENE...
eleves@eleves-VBox:~/MOUMOUN.IGOUZOULENE/projet$ ./serveur 2000
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 35048
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 35052
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 52686
Serveur. attente connexion

Commandes disponibles :
/vote question → lancer un vote
/OUI | /NON → voter
/endvote → terminer le vote (créateur)
/online → afficher les utilisateurs en li
/public message → message à tout le monde
/private nom message → message privé
/fin → quitter

[PUBLIC] uny: Soutenance le 03 juin

/online
[ONLINE] Liste utilisateurs en ligne:
1) imane
2) yassir
3) uny
/public Soutenance le 03 juin
[PUBLIC] uny: Soutenance le 03 juin
  
```

A l'aide de la commande `/private nom message`, l'utilisateur* peut envoyer un message privé à un utilisateur spécifique* :

```

eleves@eleves-VBox: ~/MOUMOUN.IGOUZOULENE...
eleves@eleves-VBox: ~/MOUMOUN.IGOUZOULENE/projet$ ./serveur 2000
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 35048
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 35052
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 52686
Serveur. attente connexion

/help
  /vote question      → lancer un vote
  /OUI | /NON         → voter
  /endvote            → terminer le vote (créateur)
  /online             → afficher les utilisateurs en li
  gne
  /public message     → message à tout le monde
  /private nom message → message privé
  /fin                → quitter

[PUBLIC] uny: Soutenance le 03 juin
[PRIVATE] imane → yassir: on doit passer à 14h

```

A l'aide de la commande `/vote question`, l'utilisateur* peut envoyer une question à laquelle les autres personnes connectées* peuvent répondre :

```

eleves@eleves-VBox: ~/MOUMOUN.IGOUZOULENE...
eleves@eleves-VBox: ~/MOUMOUN.IGOUZOULENE/projet$ ./serveur 2000
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 35048
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 35052
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 52686
Serveur. attente connexion

/help
  /vote question      → lancer un vote
  /OUI | /NON         → voter
  /endvote            → terminer le vote (créateur)
  /online             → afficher les utilisateurs en li
  gne
  /public message     → message à tout le monde
  /private nom message → message privé
  /fin                → quitter

[PUBLIC] uny: Soutenance le 03 juin
[PRIVATE] imane → yassir: on doit passer à 14h
[VOTE] avez-vous envoyé votre projet ? (OUI: 0, NON: 0)

```

Les utilisateurs ne peuvent voter qu'une seule fois* à l'aide des commandes `/OUI` ou `/NON` :

```
29 mai 15:17
eleves@eleves-VBox: ~/MOUMOUN.IGOUZOULENE...
eleves@eleves-VBox: ~/MOUMOUN.IGOUZOULENE/projet$ ./serveur 2000
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 35048
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 35052
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 52686
Serveur. attente connexion

/fin → quitter
[PUBLIC] uny: Soutenance le 03 juin
/private yassir on doit passer à 14h
Message privé envoyé à yassir.
[VOTE] avez-vous envoyé votre projet ? (OUI: 0, NON: 0)
/OUI
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 0)
/OUI
[ERREUR] Vous avez déjà voté.
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 1)

gne
/public message → message à tout le monde
/private nom message → message privé
/fin → quitter
[PUBLIC] uny: Soutenance le 03 juin
[PRIVATE] imane → yassir: on doit passer à 14h
[VOTE] avez-vous envoyé votre projet ? (OUI: 0, NON: 0)
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 0)
/NON
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 1)

/online
[ONLINE] Liste utilisateurs en ligne:
1) imane
2) yassir
3) uny
/public Soutenance le 03 juin
[PUBLIC] uny: Soutenance le 03 juin
/vote avez-vous envoyé votre projet ?
[VOTE] avez-vous envoyé votre projet ? (OUI: 0, NON: 0)
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 0)
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 1)
```

Le créateur du vote peut à tout moment mettre fin au vote à l'aide de la commande `/endvote` :

```
29 mai 15:18
eleves@eleves-VBox: ~/MOUMOUN.IGOUZOULENE...
eleves@eleves-VBox: ~/MOUMOUN.IGOUZOULENE/projet$ ./serveur 2000
[PUBLIC] uny: Soutenance le 03 juin
/private yassir on doit passer à 14h
Message privé envoyé à yassir.
[VOTE] avez-vous envoyé votre projet ? (OUI: 0, NON: 0)
/OUI
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 0)
/OUI
[ERREUR] Vous avez déjà voté.
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 1)
[PUBLIC] Serveur: [VOTE TERMINÉ] avez-vous envoyé votre projet ?
(OUI: 1, NON: 1)

/private nom message → message privé
/fin → quitter
2) yassir
3) uny
/public Soutenance le 03 juin
[PUBLIC] uny: Soutenance le 03 juin
/vote avez-vous envoyé votre projet ?
[VOTE] avez-vous envoyé votre projet ? (OUI: 0, NON: 0)
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 0)
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 1)
/endvote
[PUBLIC] Serveur: [VOTE TERMINÉ] avez-vous envoyé votre projet ?
(OUI: 1, NON: 1)
```

Les utilisateurs peuvent se déconnecter à l'aide de la commande `/fin` :

```

29 mai 15:19
eleves@eleves-VBox: ~/MOUMOUN.IGOUZOULENE...
eleves@eleves-VBox:~/MOUMOUN.IGOUZOULENE/projet$ ./serveur 2000
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 35048
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 35052
Serveur. attente connexion
Serveur. connexion recue : client adr 127.0.0.1, port 52686
Serveur. attente connexion

/private yassir on doit passer à 14h
Message privé envoyé à yassir.
[VOTE] avez-vous envoyé votre projet ? (OUI: 0, NON: 0)
/OUI
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 0)
/OUI
[ERREUR] Vous avez déjà voté.
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 1)
[PUBLIC] Serveur: [VOTE TERMINÉ] avez-vous envoyé votre projet ?
(OUI: 1, NON: 1)
/fin
eleves@eleves-VBox:~/MOUMOUN.IGOUZOULENE/projet$

/fin
→ quitter
[PUBLIC] uny: Soutenance le 03 juin
[PRIVATE] imane → yassir: on doit passer à 14h
[VOTE] avez-vous envoyé votre projet ? (OUI: 0, NON: 0)
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 0)
/NON
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 1)
[PUBLIC] Serveur: [VOTE TERMINÉ] avez-vous envoyé votre projet ?
(OUI: 1, NON: 1)
/fin
eleves@eleves-VBox:~/MOUMOUN.IGOUZOULENE/projet$

3) uny
/public Soutenance le 03 juin
[PUBLIC] uny: Soutenance le 03 juin
/vote avez-vous envoyé votre projet ?
[VOTE] avez-vous envoyé votre projet ? (OUI: 0, NON: 0)
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 0)
[VOTE] avez-vous envoyé votre projet ? (OUI: 1, NON: 1)
/endvote
[PUBLIC] Serveur: [VOTE TERMINÉ] avez-vous envoyé votre projet ?
(OUI: 1, NON: 1)
/fin
eleves@eleves-VBox:~/MOUMOUN.IGOUZOULENE/projet$

```

En exécutant l'ensemble des commandes précédentes on obtient à la fin du journal.log les informations suivantes :

```

29 mai 15:19
journal.log
~/MOUMOUN.IGOUZOULENE/projet
Ouvrir
utilisateur imane s'est connecté
utilisateur yassir s'est connecté
utilisateur uny s'est connecté
Utilisateur uny a envoyé un message public : Soutenance le 03 juin
Utilisateur imane a envoyé un message privé à utilisateur yassir : on doit passer à 14h
Utilisateur uny a lancé un vote : avez-vous envoyé votre projet ?
Utilisateur imane a voté OUI
Utilisateur yassir a voté NON
Le vote "avez-vous envoyé votre projet ?" est terminé, Le résultat : OUI=1, NON=1
Utilisateur imane s'est déconnecté
Utilisateur yassir s'est déconnecté
Utilisateur uny s'est déconnecté

```

Fonctionnement:

Ce projet s'appuie sur le corrigé du TD7 portant sur l'application multithread avec modèle dynamique. Sur cette base, plusieurs fonctionnalités ont été ajoutées afin de construire un système de messagerie enrichi avec gestion des utilisateurs et système de vote.

Fonctions ajoutées côté serveur :

- `int ajouter_identifiant(const char *id);`
→ Enregistre un nouvel identifiant utilisateur s'il n'est pas déjà utilisé.
(La fonction renvoie 0 si l'identifiant existe déjà)
- `int supprimer_identifiant(const char *id);`
→ Supprime un identifiant de la liste des utilisateurs connectés à la déconnexion.
- `void listeOnline(int canal);`
→ Envoie au client la liste des utilisateurs actuellement connectés.
- `void envoyerMessagePublic(const char *expediteur, const char *message);`
→ Diffuse un message à l'ensemble des utilisateurs connectés (chat public).
- `int envoyerMessagePrive(const char *expediteur, const char *destinataire, const char *message, int canal_src);`
→ Envoie un message privé à un utilisateur spécifique, si celui-ci est connecté.
(La fonction renvoie 0 si le destinataire n'est pas connecté)
- `void resultatVote(void);`
→ Affiche à tous les utilisateurs les résultats actuels d'un vote en cours (nombre de votes OUI/NON).

Variables globales associées :

- `char *identifiants[MAX_CLIENTS];`
→ Tableau contenant les identifiants des utilisateurs connectés.
- `int canaux_clients[MAX_CLIENTS];`
→ Canaux de communication associés à chaque utilisateur.
- `char vote_question[], char vote_createur[], int vote_oui, int vote_non, int vote_en_cours, int deja_vote[];`
→ Variables nécessaires à la gestion d'un vote unique (contenu de la question, initiateur, décompte des votes et suivi des utilisateurs ayant déjà voté).

Fonction ajoutée côté client :

- `void *reception(void *arg);`
→ Fonction thread dédiée à la réception des messages envoyés par le serveur.