

# Rapport de projet : Chat201 – édition thread & réseau

Daniel DEFOING (ddef0003)      Belinda ÖZNUR (bozn0003)  
Haluk YILMAZ (hyil0003)

14 décembre 2024

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Choix du langage :pourquoi C++ plutôt que C ?</b>	<b>1</b>
<b>3</b>	<b>Visualisation de l’architecture du programme</b>	<b>1</b>
3.1	Le serveur en tant que point relais des clients . . . . .	1
3.2	Conceptions communes aux clients et au serveur . . . . .	2
3.3	Conception des clients . . . . .	2
3.4	Conception du serveur . . . . .	2
<b>4</b>	<b>Difficultés rencontrées et solutions trouvées</b>	<b>2</b>
4.1	Problèmes de synchronisation . . . . .	2
4.2	Garantie de l’intégrité du contenu partagé par les clients . . . . .	2

## 1 Introduction

Ce rapport décrit globalement la conception du second projet dans le cadre du cours de Systèmes d'Exploitation (INFO-F201). Il présente les choix de conception qui ont guidé notre développement, et les difficultés qui ont pu survenir durant celui-ci. Pour voir de façon précise les changements ayant eu lieu tout au long du projet et la contribution de chacun, veuillez consulter le repository GitHub de notre projet.

## 2 Choix du langage : pourquoi C++ plutôt que C ?

Une raison fondamentale qui a guidé cette décision est que C++ possède des fonctionnalités absentes en C tels que les références, les chaînes de caractères (ou *strings*) ou encore les classes. Or, dans le cadre du projet présenté ici, ces éléments apportent une plusvalue non négligeable :

— Discuter des avantages du C++ (il y en a beaucoup)...

Ceci montre que des avantages majeurs ont été trouvés à utiliser C++ plutôt que C, alors qu'aucun avantage n'a été identifié en faveur de l'utilisation de C par rapport à C++. Ce choix du langage s'est donc naturellement imposé comme le plus adapté.

## 3 Visualisation de l'architecture du programme

Cette section discute des choix de conception du programme final, des considérations qui permettront au lecteur de mieux comprendre la nature de ces choix et des problèmes rencontrés qui en ont suivi.

### 3.1 Le serveur en tant que point relais des clients

La structure d'un programme client-serveur tel que celui présenté ici peut être visualisée très simplement sous la forme d'un Graphe Étoile [1], avec le serveur au centre et les clients aux extrémités de chaque branche.

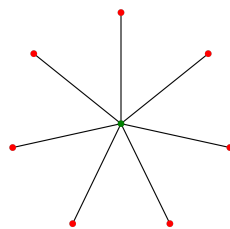


FIGURE 1 – *Graphe Étoile typique.*

Cette représentation montre bien qu'un seul serveur se charge de *relayer* les informations à passer d'un client à un autre ou d'un client vers lui-même (confirmation de (dé)connexion notamment).

TODO : discuter du TCP/IP brièvement

Dans la suite, nous discuterons en profondeur de la façon dont les noeuds de ce graphe communiquent.

### 3.2 Conceptions communes aux clients et au serveur

- 2 Queues à vider (par un nombre variables de threads selon qu'on est dans le client ou le serveur)
- Méthodes d'envoi des messages : utilisation d'une classe `Message`
- usage de `std::string` la plupart du temps

### 3.3 Conception des clients

- 2 threads, 2 Queues (écho avec la section "Conceptions communes")
- Gestion de la connexion
- Gestion de la déconnexion (quel thread tue l'autre, comment on se reconnecte)
- ...

### 3.4 Conception du serveur

- Plusieurs threads, 2 Queues (écho avec la section "Conceptions communes") : accès concurrents aux Queues, problème de producteur-consommateur
- Gestion des (dé)connexions
- Mapping des utilisateurs à leur file descriptors et inversement
- Gestion des sockets : surveillance permanente via `poll` (ou `epoll` le cas échéant)
- ...

## 4 Difficultés rencontrées et solutions trouvées

### 4.1 Problèmes de synchronisation

- côté serveur, accès concurrents (problème producteur-consommateur)
- Signaux asynchrones (mentionner la conception du `SignalManager` dans l'explication de la solution)
- ...

### 4.2 Garantie de l'intégrité du contenu partagé par les clients

- Gestion des tailles limites (longueur de pseudos et de messages)
- Comment s'est-on assurés que le messages étaient bien transmis sans perte ?

## Références

- [1] Wikipedia. (2019, 21 janvier). *Graphe étoile*. URL : [https://fr.wikipedia.org/wiki/Graphe\\_%C3%A9toile](https://fr.wikipedia.org/wiki/Graphe_%C3%A9toile), consulté le 12/12/2024.