

Une application de messagerie instantanée



Un projet FAR de programmation en C en cycle itératif par

CHEKOU Nesrine, GUYON Valentin & VEBER Mathieu

ANNEE 2018-2019

Responsable : M. TIBERMACINE

Sommaire

[Itération 1 2](#_30j0zll)

[1.](#_1fob9te) Les difficultés rencontrées 2

[2.](#_2et92p0) La répartition du travail 3

[3.](#_tyjcwt) Compilation et exécution 4

[4.](#_3dy6vkm) Le schéma du protocole de communication 5

# Itération 1

## Les difficultés rencontrées

La principale difficulté rencontrée fut l’utilisation du multi-threading. De nouveau, ne pas avoir d’IUT ou BTS informatique dans notre groupe fut contraignant. Nous nous sommes heurté à de nombreuses erreurs spécifique au langage C et avons perdu de longues heures de débogage.

## La répartition du travail

Pour ce rendu, nous avons de nouveau travaillé ensemble pour coder. On souligne toutefois que Valentin s’est davantage concentré sur le programme client, Mathieu s’est consacré au server et Nesrine a contribué à parts égales aux deux programmes.

La rédaction de ce rapport ainsi que l’élaboration du graphe reviennent principalement à Valentin et Nesrine.

## Compilation et exécution

Pour compiler les programmes, place vous à l’aide du terminal dans le répertoire où se trouve les fichiers server.c et client.c, puis taper dans le terminal les 3 commandes suivantes :

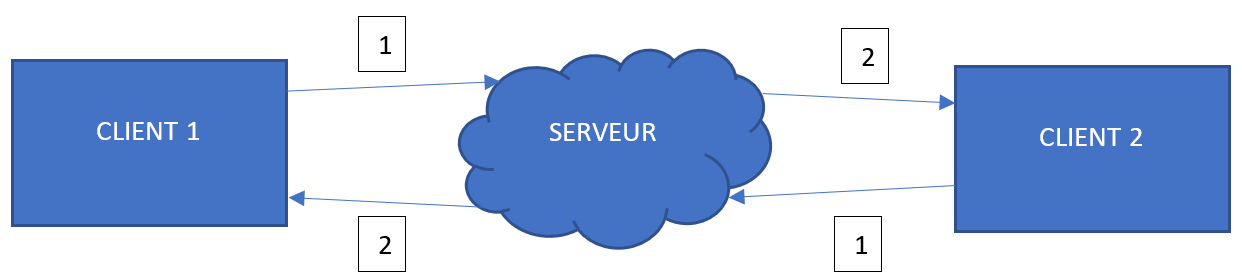
* gcc -pthread -o server server.c
* gcc -pthread -o client1 client.c
* gcc -pthread -o client2 client.c

Pour exécuter les programmes taper les commandes suivantes dans l’ordre, pour obtenir l’<ip> faites un ifconfig et récupérer votre adresse ipv4 car c’est cette dernière qui correspond à l’<ip> :

* ./server
* ./client1 <ip>
* ./client2 <ip>

Voilà, vous pouvez maintenant chatter avec vous-même !

## Le schéma du protocole de communication



Deux threads côté client. Le premier réceptionne les messages relayés par le serveur et les affiche à l’utilisateur. Le second récupère le message saisi par l’utilisateur et l’envoie au serveur.

Deux threads également côté serveur. Un chargé de la transmission du client 1 au client 2 et le second du client 2 au client 1.

A présent, un client peut envoyer des messages au serveur même sans réponse de son interlocuteur, et il peut recevoir plusieurs messages sans avoir à y répondre.