

Réseaux : modèles, protocoles, programmation

Université Paris 8 – Vincennes à Saint-Denis
UFR MITSIC / L2 informatique

Séance b (TP) : Blablabla

N'oubliez pas :

- Les TPs doivent être rendus par courriel au plus tard la veille de la séance suivante avec “[rmpp]” suivi du numéro de la séance et de votre nom dans le sujet du mail, par exemple “[rmpp] TPb Rauzy”.
- Quand un exercice demande des réponses qui ne sont pas du code, vous les mettez dans un fichier texte **reponses.txt** à rendre avec le code.
- Le TP doit être rendu dans une archive, par exemple un tar gzippé obtenu avec la commande `tar czvf NOM.tgz NOM`, où **NOM** est le nom du répertoire dans lequel il y a votre code (idéalement, votre nom de famille et le numéro de la séance, par exemple “rauzy-tpb”).
- Si l'archive est lourde (> 1 Mo), merci d'utiliser <https://bigfiles.univ-paris8.fr/>.
- Les fichiers temporaires (si il y en a) doivent être supprimés avant de créer l'archive.
- Le code doit être proprement indenté et les variables, fonctions, constantes, etc. correctement nommées, en respectant des conventions cohérentes.
- Le code est de préférence en anglais, les commentaires (si besoin) en français ou anglais, en restant cohérent.
- **N'hésitez jamais à chercher de la documentation par vous-même sur le net!**

Dans ce TP :

- Écriture d'un client pour un protocole de tchat simple.

Exercice 0.

Le protocole de tchat.

1. Tant qu'un client n'est pas identifié le serveur lui envoie **WHO** en réponse à chaque commande, sauf à la commande d'identification.
La commande d'identification est **IAM** suivi d'une espace suivi d'un pseudo.
Si le pseudo contient une espace ou si il est déjà utilisé par quelqu'un d'autre, le serveur répond avec **ERR**.
Dans le cas contraire, si tout va bien, le serveur répond **ARV** suivi d'une espace suivi du pseudo donné avec la commande d'identification.
Le serveur prévient avec le même message les autres clients déjà identifiés de l'arrivée du nouveau participant.
Le client dispose de 15 secondes pour s'identifier une fois connecté. Passé ce délais, le serveur coupe la connexion.
2. Une fois identifié, la liste des commandes existantes est la suivante :
 - En réponse à la commande **MSG** suivi d'une espace suivi d'un texte jusqu'à une fin de ligne (`\n`), le serveur broadcast à tous les clients identifiés le message **MSG** suivi d'une espace suivi du pseudo de l'émetteur suivi du texte.
 - En réponse à la commande **PRV** suivi d'une espace suivi d'un pseudo suivi d'une espace suivi d'un texte jusqu'à une fin de ligne (`\n`), le serveur répond **ACK** à l'émetteur et le message **PRV** suivi d'une espace suivi du pseudo de l'émetteur suivi du texte à destinataire du message privé (c'est à dire le client identifié avec le pseudo donné par l'émetteur).
Si le pseudo ne correspond pas à celui d'un client identifié, le serveur répond **ERR** à l'émetteur.
 - En réponse à la commande **WHO** le serveur répond à l'émetteur **LST** suivi d'une espace suivi du nombre de clients identifiés suivi d'un retour à la ligne (`\n`) suivi des pseudos des clients identifiés un par ligne.
 - En réponse à la commande **BYE**, le serveur répond à l'ensemble des clients identifiés avec le message **LFT** suivi d'une espace suivi du pseudo de l'émetteur.
 - En réponse à la commande **TIC**, le serveur répond **TOC** à l'émetteur. Les clients identifiés **doivent** envoyer des **TIC** toutes les 15 secondes maximum, sinon ils sont déconnectés et leur départ est notifiés aux autres clients identifiés avec **OUT** suivi d'une espace suivi de leur pseudo.
3. Un serveur de test tourne sur l'hôte **pablo.rauzy.name** sur le port **4567**. Il est relancé automatiquement toutes les 30 minutes (vers pile et vers 30) pour assurer son fonctionnement sans trop de surveillance.
4. À vous d'écrire un client convivial pour ce tchat!