Pablo Rauzy
pablo.rauzy@univ-paris8.fr
https://pablo.rauzy.name/teaching/rmpp

UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

Réseaux : modèles, protocoles, programmation

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

Séance b (TP) : Jeu des allumettes

N'oubliez pas :

- Les TPs doivent être rendus par courriel au plus tard le lendemain du jour où ils ont lieu 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 mettrez 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 :

- Compréhension de programme.
- Appréhender la conception d'un protocole.

Exercice 0.

Bien démarrer.

1. Pensez à organiser correctement votre espace de travail, par exemple tout ce qui se passe dans ce TP pourrait être dans `~/rmpp/sb-tp/`.
2. Pensez à nommer le dossier que vous rendrez avec votre nom. Si vous ne le faites pas tout de suite, pensez à le faire avant de rendre votre TP.
3. Dans ce TP vous n'avez pas à écrire de code, mais à en comprendre. Comme d'habitude, n'hésitez pas à consulter les pages de `man` ou le web pour vous documenter.
4. Vous sont fournis le code d'un serveur de jeu des allumettes à plusieurs, écrit en C, et le code d'un client pour ce jeu écrit en Python *quick and dirty*.

   Le serveur peut être compilé simplement avec `gcc *.c`.

   Le client se lance avec `python3`.

   → Compilez le serveur, lancez le et jouer un peu avec (à plusieurs si possible). Les commandes utilisables dans le client sont trouvables dans sa fonction `waiting_usr`.

Exercice 1.

Architecture du serveur.

1. → Expliquez les rôles de chacun des modules du serveur.
2. → Justifiez tant que vous pouvez les choix architecturaux (organisation du code, des structures de données, des headers et de ce qui est ou pas dedans).

Exercice 2.

Protocole.

1. → Explicitez le protocole utilisé par le client et le serveur pour communiquer. En plus de fouiller le code, n'hésitez pas à utiliser des outils comme `nc` pour tester vos hypothèses.
2. → En vous aidant du code et du protocole, déduisez les règles du jeu.
3. → Le client implémente-t-il tout ce qui est prévu par le serveur ? Détaillez.