

**TP n° 3 - 2<sup>ème</sup> année - RSA Réseaux**  
 2015-2016  
**TP Programmation Avancée**

L'ensemble des sources dont vous avez besoin pour votre TP sont à votre disposition à l'URL suivante : <http://www.loria.fr/~ichris/Templates.tar.gz>. Le répertoire Templates est composé de 4 sous-répertoires :

- Select
- Raw
- Urgent
- IPv4

Le TP se déroulera en deux séances (la deuxième séance est facultative). A la fin de chaque séance, vous enverrez à votre encadrant les exercices terminés et vous supprimerez localement les fichiers que vous avez écrits.

#### Exercice 1

Dans le répertoire **Raw** se trouvent les sources d'un client UDP qui envoie l'heure à un serveur, attend un message du serveur et quitte l'application. Développez le serveur qui affiche l'heure reçue, renvoie la donnée reçue et se met ensuite en attente sur la requête d'un autre client.

#### Exercice 2

L'objectif de cet exercice est de vous faire manipuler le concept d'entrées/sorties multiplexées.

- Tester tout d'abord la version du serveur echo multi-client : **Select/servmulti\_tcp.c**. Pour la partie client, vous pouvez utiliser **telnet**.
- Modifier cette version pour en faire un serveur multi-clients utilisant la primitive **select()**. Il s'agit d'implanter les notions vues en TD.

#### Exercice 3

L'objectif de cet exercice est de vous faire manipuler le concept de socket raw.

- Tout d'abord copier le fichier **Raw/icmpd-incomplet.c** dans **icmpd.c** et complétez-le
- Tester les différents scénarios permettant l'arrivée de messages ICMP de type ICMP\_ECHOREPLY, ICMP\_UNREACH et ICMP\_TIMXCEED. Vous pourrez également vérifier la réception des messages ICMP via **Wireshark**.

#### Exercice 4

L'objectif de cet exercice est de vous faire manipuler le concept de donnée urgente appelée aussi message OOB (Out Of Band).

- Tout d'abord copier le fichier **Urgent/serverUrgentTCP-incomplet.c** dans **serverUrgentTCP.c** et complétez-le.
- Exécuter le serveur en connectant le client dont le fichier source est **Urgent/clientUrgentTCP.c**. Analyser et expliquer le comportement du serveur.
- Observer avec **Wireshark** l'en-tête TCP et notamment les informations relatives aux données urgentes.

#### Exercice 5

L'objectif de cet exercice est de vous faire manipuler le concept de multicast/communication de groupe.

- Ecrire le code de l'émetteur **sender.c** qui enverra la date courante à une adresse multicast. Il suffit pour cela d'adapter le code source **Raw/clientUDP.c**.
- Ecrire le code du récepteur qui affiche la date courante reçue sur cette adresse multicast.
- Tester votre application en lançant plusieurs récepteurs sur une même machine (voire sur d'autres machines).