

## PR6 – Programmation réseaux

### TP n° 8 : Un émetteur QOTD

À l'époque des ordinateurs partagés, il était habituel qu'une citation amusante, le *quote of the day* (QOTD) apparaisse lors du *login*. Avec l'arrivée de l'Internet, on a développé le protocole QOTD, qui permet à la base de données de citations de résider sur un serveur central — lors du *login*, un client QOTD consulte un serveur central qui lui sert une citation à afficher.

Le protocole QOTD, défini par la RFC 865, est extrêmement simple. Le client envoie un datagramme UDP vide au port 17 du serveur. Le serveur répond par un datagramme UDP contenant une chaîne de caractères ne dépassant pas 512 octets.

1. Écrivez un serveur UDP en langage C qui écoute sur le port 1717. Lorsqu'il reçoit un datagramme, quel que soit son contenu, il répond par un datagramme contenant votre chaîne de caractères préférée. Pour répondre il suffit qu'il utilise les caractéristiques du client qui se trouvent dans le paquet envoyé par le client. En fait, ici vous ne connaissez jamais le port utilisé côté client mais vos programmes l'utilisent. Testez votre serveur avec la commande `nc` et l'option `-u` pour préciser qu'il s'agit d'UDP.
2. Écrire un client Java `Client.java` tel que `java Client n ip_adress` envoie `n` fois un datagramme UDP vide au port 1717 du serveur d'adresse IPv4 `ip_adress` et après chaque envoi il attend un message UDP sur la socket utilisée pour l'envoi du paquet vide et affiche son contenu. Testez votre client Java avec votre serveur C.
3. Un fichier `fortune` contient une liste de citations séparées par des lignes contenant un signe pourcent « % ». Modifiez votre programme pour qu'il tire une citation au hasard dans un fichier au format fortune. Vous trouverez une collection de fortunes dans le répertoire `/usr/share/games/fortunes` de l'UFR.
4. Programmez le client précédent en C et le serveur précédent en Java.