

Partie I :

1)

la structure headerPDU une valeur caractère pour le type de données, ainsi qu'une chaîne de caractère de taille BUFSIZ pour la taille du message

La structure dataPDU contient une chaîne de caractère avec les données

La structure messagePDU contient les deux structure précédente pour regrouper la entête du message, avec les infos, ainsi que son contenu.

2)

a) La longueur des données permet de vérifier si on a bien reçus le message intégralement

b)

c) On fait deux écritures pour que le serveur puisse vérifié que la taille attendu correspond bien à la taille à l'arrivé

3) La fonction setsockopt permet d'écrire les options de la socket :

le premier paramètre correspond au socket que l'on veut fixer

le deuxième paramètre correspond au niveau de l'option que l'on veut fixer, ici level prendra donc comme valeur le numéro de protocole TCP

le troisième paramètre est utilisé pour déterminer l'option pour setsockopt(). Ici on utilise TCP_NODELAY pour forcer l'envoi immédiat des données contenues dans la socket, quelle que soit la taille du paquet

le quatrième paramètre est un paramètre resultat contenant initialement la taille du tampon pointé par optval, et rempli en retour pour indiquer la taille effective des valeurs renvoyées. Si aucune option n'est fournie ou renvoyée, optval peut être NULL.

5) INADDR_ANY est un numéro fictif qui est la pour indiquer que le serveur attend des demandes sur n'importe qu'elle adresse IP.

Port est initialisé à 0 pour avoir un port inutilisé à la base et permettre une allocation dynamique

6) On utilise cet appel système pour récupérer l'extrémité local et donner l'adresse ip local comme on a fait usage de INADDR_ANY.