

Algorithmique parallèle et distribuée

Socket UDP et Multicast

Un premier client/serveur

Exercice 1

A partir des squelettes présentés en cours, développez une première application client/serveur UDP, telle que le serveur :

- renvoie chaque message reçu d'un client ;
- y ajoute le préfixe "RECEIVED FROM XXX.XXX.XXX.XXX : " (ou XXX.XXX.XXX.XXX désigne l'adresse de la machine émettrice).

Quelle heure est il ?

Exercice 2

Un grand classique : développez une application client/serveur UDP, de telle sorte que chaque fois que le serveur reçoit un paquet, il retourne l'heure courante à l'émetteur de ce paquet.

Quelques remarques :

- le paquet émis par le client peut être vide, de toute façon le serveur ne tient pas compte de son contenu ;
- pour connaître l'heure courante, reportez vous sur les classes *Date* et *Calendar* ;
- faut il modifier quelque chose pour que le serveur s'occupe de plusieurs clients ?

Un premier chat

Exercice 3

Écrire une petite application utilisant une socket UDP qui permet à un utilisateur d'envoyer et de recevoir des messages en interaction avec une machine distante.

Si l'on y réfléchit bien, le programme doit à la fois permettre d'envoyer et de recevoir des messages, mais la méthode *receive* est bloquante... Alors comment faire ?

Pour la saisie au clavier, reportez vous sur la classe *Scanner* ;

Un forum de discussion

Exercice 4

Ecrire une petite application utilisant une socket de multicast UDP qui :

- envoie vers un port et une adresse (passés en paramètre) d'un groupe de multicast ce qui est écrit depuis le clavier ;
- affiche à l'écran tout ce qui est reçu sur ce groupe de multicast à destination de cette adresse et de ce port.

Pensez à vous inspirer de l'exercice précédent, le principe est très similaire.

Essayez ensuite d'utiliser tous la même IP ainsi que le même numéro de port...