

M4102C : Programmation répartie

TP4 : Programmation RPC

Objectif :

Mise en œuvre du mécanisme simple de RPC du système **Linux**.

Travail demandé :

Développer en langage **C**, une application répartie mettant en œuvre le mécanisme de RPC avec utilisation de XDR (eXternal Data Representation) et mise en œuvre de la diffusion.

Spécifications :

1- Ecrire un programme C qui fait la somme des nombres entiers appartenant à un intervalle [x,y] donné par l'utilisateur.

2- Transformer ce programme pour faire calculer la somme par une autre machine

- problème pour transférer deux paramètres
- solution : envoyer une structure contenant les 2 params
- utiliser XDR
- décrire la structure dans le fichier .x

```
struct param { int borne_inf;  
               int borne_sup; };
```

3- transformer le programme pour faire calculer la somme par plusieurs machines (pour comparer leurs vitesses respectives par exemple)

- 3.1- en donnant le nom de chaque machine
- 3.2- en faisant une diffusion
- utiliser clnt_broadcast à la place de callrpc().

clnt_broadcast : Le RPC est diffusé à toutes les machines connectées sur le même réseau en utilisant l'adresse de broadcast

Format :

```
enum clnt_stat  clnt_broadcast(u_long prognum, u_long versnum,  
u_long procnum, xdrproc_t inproc, char * in, xdrproc_t outproc,  
char * out, resultproc_t eachresult);
```

Arguments :

Prognum : Le numéro de programme associé à la procédure distante (exemple : SOMMEPROG = 0x20000001).

Versnum : Le numéro de version associé à la procédure distante (exemple : SOMMEVERS = 1).

Procnum : Le numéro de procédure associé à la procédure distante (exemple SOMME_INTERVALLE = 1).

Inproc : La procédure XDR utilisée pour coder les arguments de la RP.

In : Un pointeur vers les arguments de la RP.

Outproc : La procédure XDR utilisée pour décoder le résultat de la RP.

Out : Un pointeur vers le résultat de la RP.

Eachresult : Fonction appelée à chaque fois que l'on reçoit une réponse.

Elle est déclarée comme suit :

```
int eachresult(char *resultsp, struct sockaddr_in *addr)
```

resultsp est un pointeur sur le résultat de la RP.

addr est un pointeur vers une structure "sockaddr_in" qui contient l'@ de l'hôte qui a répondu.

La fonction doit renvoyer 0 (faux) si on veut attendre d'autres résultats.

Si eachresult est NULL, la fonction clnt_broadcast n'attend pas de réponse.