

Programmation Distribuée 2009

TP 1 : Appel de procédure à distance (RPC)

http://cafeteria.u-strasbg.fr/~gancars/TP_ProgDistribuee/

Exercice 1

Compiler puis tester l'exemple se trouvant dans le répertoire `PremierExemple`. Utiliser la commande `makefile`. Constater que votre serveur est enregistré au niveau du serveur de liaison avec la commande : `/usr/sbin/rpcinfo -p`. Le numéro de programme en décimal de votre service est 536871168.

L'affichage sur un PC:

```
client: variable n (debut) : 1094861636 DCBA,
client: variable n (fin) : 1094861636,
client: variable res : 1094861637
```

⇒ L'affichage de la première ligne diffère sur une SPARC (Machine SUN) :

```
client: variable n (debut) : 1094861636 ABCD,
```

Affichage :

```
serveur: variable n (debut) : 1094861636,
serveur: variable n (fin) : 1094861637,
serveur: variable res : 1094861637
```

Questions :

- 1.1) Dans le serveur pourquoi déclarer la variable 'res' static ?
- 1.2) Expliquez pourquoi l'affichage au niveau du client est différent selon que l'on utilise une SPARC ou un PC.
- 1.3) Pourquoi la valeur finale de (*n) est différente sur le serveur et sur le client ?
- 1.4) Dans le Makefile, expliquer la ligne de compilation `gcc -Wall -o client client.c -lrpcsvc -lnsl`

Exercice 2

Définir un appel RPC qui permette de calculer, de façon distante, la division entière d'un nombre par un autre. Le serveur de calcul (qui est responsable de la division) se trouvera sur votre machine. Vous exécuterez le client depuis une autre machine. Pour cela, il est possible de garder en permanence une fenêtre sur cette autre machine avec la commande `ssh votreLogin@machineDuVoisin`. Vous éditez, compilerez les fichiers sources sur votre machine. Les sources pour débiter cet exercice se trouvent dans le répertoire `DiviseExercice`.

ATTENTION: ces fichiers sources ne sont pas "corrects". Il faut remplacer les "???" par du code C.

Expliquer la ligne de compilation `gcc -o serveur serveur.o xdr_entiers2.o -lnsl`

Exercice 3

On désire implanter un serveur de calcul sur des matrices 2X2 à coefficients réels qui offre comme service la multiplication de deux matrices. Mettre en place un tel serveur sur votre machine.

Exercice 4

Ajouter le service : addition de deux matrices. Le client offrira la possibilité d'appeler le service de multiplication ou d'addition, au choix.

Exercice 5

Reprenez les exercices 4 et 5 en considérant que les matrices à multiplier peuvent être NxN. Mettez en évidence qu'il existe un time-out (au niveau du client) et une retransmission lorsque l'appel de fonction distant prend un peu de temps.

Rappel

* Enregistrement d'un service :

```
int registerrpc(u_long prognum, u_long versnum, u_long procnum, char *(*procname)(), xdrproc_t inproc, xdrproc_t outproc);
```

* Appel d'un service :

```
int callrpc(char *host, u_long prognum, u_long versnum, u_long procnum, xdrproc_t inproc, char *in, xdrproc_t outproc, char *out);
```

* Mise en attente d'un serveur :

```
svc_run()
```

* Utiliser les pages de manuel : `man registerrpc`, `man xdr_string`. Regarder les fichiers `rpc.h` et `xdr.h` du répertoire `/usr/include/rpc`