# TP noté – RPC/RMI

ENSIIE

Middleware

Cédric Wemmert

# Exercice 1 – RPC

Définir un appel RPC qui permette de trier, de façon distante, un tableau d'entiers passé en paramètre.

### Rappels

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()

Utilisez les pages de manuel : man registerrpc, man xdr\_string. Regardez les fichiers rpc.h et xdr.h du répertoire /usr/include/rpc.

# Exercice 2 – RMI

Nous disposons de services de calcul numérique implantés sous la forme d’un objet Calculator. Cet objet permet d’utiliser les méthodes : add, sub, mul, div. On souhaite pouvoir rendre cet objet accessible à distance via Java RMI.

public class Calculator{

public Calculator() { }

public long add(long a, long b){ return a + b; }

public long sub(long a, long b){ return a - b; }

public long mul(long a, long b){ return a \* b; }

public long div(long a, long b){ return a / b; }

}

1. On souhaite rendre chacune de ces méthodes accessibles à distance. Donnez alors la structure des interfaces qui seront partagées par le serveur et le client. Complétez le code de ces méthodes afin qu’elles puissent gérer les erreurs dues à leur appel à distance.

2. Ecrivez le programme serveur permettant de rendre accessible à distance l’ensemble de ces méthodes.

3. Ecrivez un programme client qui effectue différents calculs à distance.