

SRCS : Systèmes Répartis Client/Serveur

TD 04 – Étude du protocole RPC et d'Inetd

Janvier 2016

Exercice 1 : Sémantique des appels RPC

Question 1

Quel mécanisme de base des langages de programmation le RPC étend-il ?

Question 2

Comparer les mécanismes de passage de paramètres de ce mécanisme de base et du RPC ?

Question 3

On est parfois amené à émettre plusieurs fois les messages d'appels et/ou de retour associés au RPC. Pourquoi ? Quelles en sont les conséquences ?

Question 4

A quoi correspondent les sémantiques d'exécution au plus une fois, au moins une fois et exactement une fois pour un RPC ? Pourquoi ne les retrouvent-on pas dans le mécanisme de base de la question 1 ?

Exercice 2 : Extension : RPC redondant**Question 1**

On se place dans le cas où un appel distant active non pas une instance de services, mais plusieurs simultanément. Proposer des situations pour lesquelles une telle redondance peut être souhaitée.

Question 2

On se place dans la situation où les instances de services activées sont identiques (les services invoqués sont implantés de la même façon). On « s'attend » donc à ce que les résultats fournis par ces exécutions soient identiques. Quelles hypothèses générales doit-on néanmoins faire pour que cela soit le cas ?

Question 3

Deux clients invoquent le même service sur un même groupe de serveurs. Concernant l'ordre de réception des requêtes, que peut-il se passer ? Proposer une solution pour résoudre le problème.

Exercice 3 : Étude du protocole RPC

Le but de cet exercice est d'étudier les mécanismes internes du protocole RPC de Sun. Une telle étude se révèle nécessaire dès lors que l'on a à réaliser une implantation de ce protocole. Pour cela on se base sur la RFC 1831 (extrait fournit en annexe) qui décrit son fonctionnement.

Question 1

Quels sont les types de messages définis ?

Question 2

Un message RPC commence toujours par un champ xid. A quoi sert-il ? Pourrait-on le remplacer par l'adresse IP du client ?

Question 3

Le corps d'un message d'appel contient les champs prog , vers et proc. A quoi correspondent-ils ?

Question 4

Le protocole définit qu'un appel peut être accepté sans toute fois aboutir à un succès. Quels sont les cas possibles et qu'a-t-on comme indication dans ces cas ?

Exercice 4 : Fonctionnement de l'Inetd

Les serveurs sur une machine sont des processus qui fonctionnent en permanence. Le but de cet exercice est d'étudier un serveur de serveur (*inetd*).

Question 1

Quels sont les intérêts d'un serveur de serveurs ?

Question 2

Inetd doit lancer des processus en fonction des requêtes qu'il reçoit. Comment rediriger la socket vers le nouveau processus ? Qu'est-ce que cela implique pour l'écriture d'un serveur ?

Question 3

A l'aide de la primitive `select()`, schématiser le fonctionnement d'inetd.

Question 4

On considère qu'un fichier (`/etc/inetd.conf`) contient la liste des serveurs que inetd représente. Quelles sont les informations nécessaires à ce fichier ?

Exercice 5 : XDR

Le but de cet exercice est d'illustrer le mécanisme de représentation de données XDR. Pour cela on se base sur la RFC 1014 qui est fournie en annexe.

Question 1

À la fin de la RFC (page 7, juste avant la section 7), vous trouverez un exercice :

Suppose now that there is a user named "john" who wants to store his lisp program "sillyprog" that contains just the data "(quit)". His file would be encoded as follows :

Répondre à cet exercice sans donner les codes hexadécimaux des caractères alphabétiques.