

Systeme de communication Client-Serveur

Touil Doniez
2 ING 02



Le cadre général

Un serveur attend des questions de clients dans un tube fifo1.

Une question correspond à la demande d'envoi de **n** nombres tirés au sort par le serveur (**n** est un nombre aléatoire compris entre **1** et **NMAX** tiré au sort par le client).

Dans sa **question**, le client envoie également son **numéro (PID)** de telle sorte que le serveur puisse le réveiller par l'intermédiaire du **signal SIGUSR1** quand il a écrit la **réponse**.

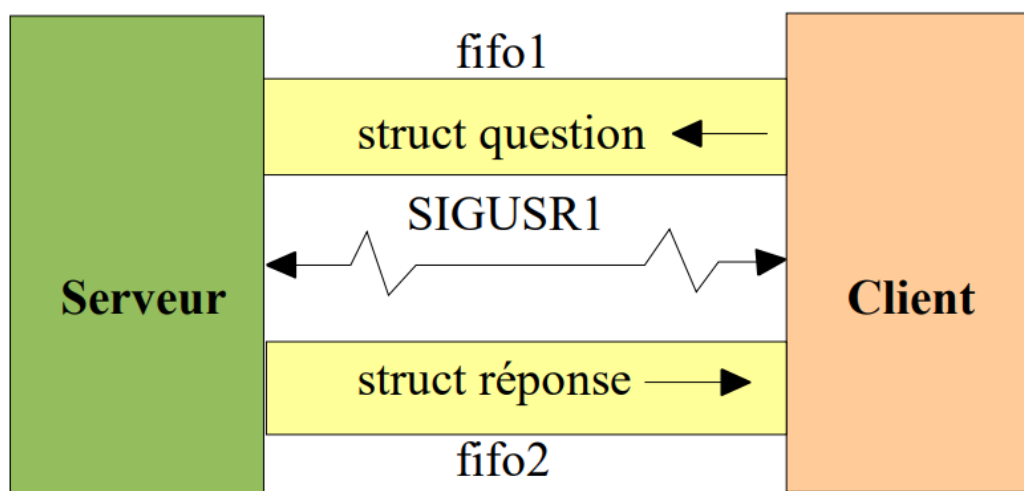
En effet, **plusieurs clients** pouvant être en attente de réponses dans le **même tube**, il est nécessaire de définir un protocole assurant que chaque client lit les réponses qui lui sont destinées.

Le client avertit par ce même signal le serveur quand il a lu les réponses.

⇒ Démarche :

Le serveur va s'ouvrir tout en créant 2 tubes nommées et reste en attente d'une question de la part du client le client va fournir un question tout en envoyant un signal pour faire réveiller le serveur qu'une question a été écrite dans le tube le serveur lui-même va lire le question pour fournir une réponse et la écrit dans l'autre tube tout en envoyant un signal indiquant que la réponse a été écrite dans le tube. Les signaux qui ne sont pas SIGUSR1 vont causé le déclenchement du handler du fin de serveur. Sachant que le signal SIGUSR1 prend les numéros 10, 30 et 16.

Conception



Il existe 2 acteurs principal :

- Un client
- Un serveur

Le but du programme c'est d'avoir une communication entre le client et le serveur via 2 tubes nommés

Identification des acteurs :

Client : Il va fournir une question qui est compose par son pid et un nombre n aléatoire. Ce client va écrire sa réponse dans un premier tube nommés et va lire la réponse d'après un deuxième tube.

Fifo 1: accès par écriture seulement.

Fifo2: accès par lecture seulement.

Serveur: Il va fournir une réponse d'après l'entier lu du fifo1 la réponse est un tableau de n nombre aléatoires et le pid du serveur. Le serveur va être réveillé par un signal SIGUSR1 envoyé de la part du client lors d'écriture d'un question dans fifo1.

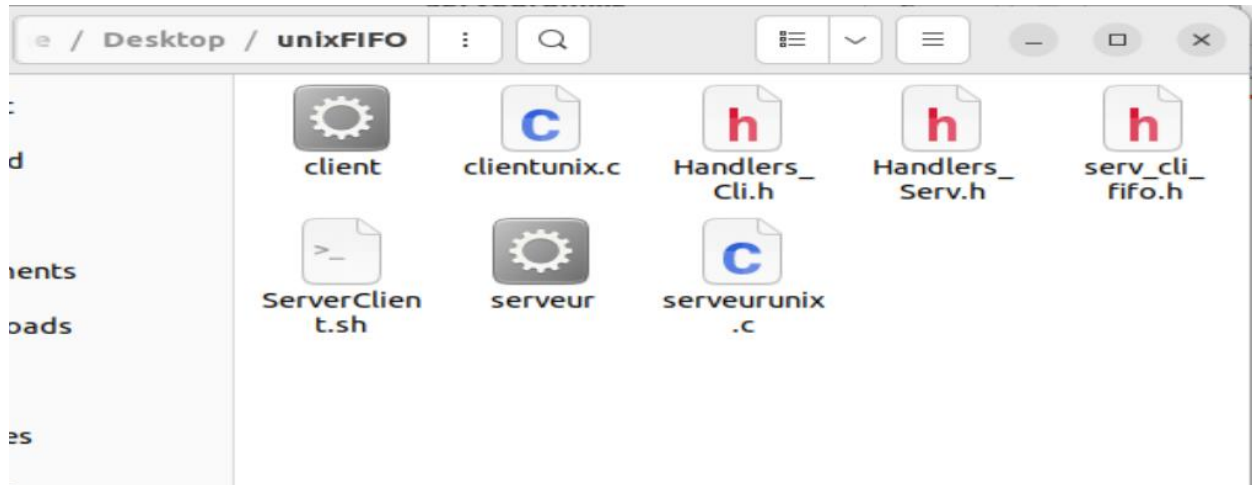
La réponse est mise dans fifo2 .

Fifo1: accès par lecture seulement.

Fifo2: accès par écriture seulement.

La réalisation

La structure de mini projet



Chaque fichier est bien commenté pour une bonne compréhension de code.

Un fichier « serv_cli_fifo » qui contient la définition des structures de réponse et question ainsi que les constantes et la déclaration des tubes nommés

Un fichier « Handlers_serv.h » qui contient le handler de signal SIGUSR1 et le handler qui traite la fin du serveur.

Un fichier « Handler_cli.h » qui contient le handler de signal SIGUSR1.

Un fichier « clientunix.c » c'est le programme principal de coté client.

Un fichier « serveurunix.c » c'est le programme principal de coté seueur.

Un fichier « ServerClient.sh » qui représente le syntaxe du bash scripting language pour une exécution automatique du projet.

Deux fichier exécutable ont été créée par les commandes shell :

Gcc clientunix.c -o client

Gcc serveurunix.c -o serveur

Clôture

La gestion d'une application peut se faire en client-serveur en utilisant des tubes nommés comme moyen de communication interprocessus. Un tube dont le nom est connu des client reçoit des requêtes. Le serveur fournit la réponse en créant un tube pour chaque client. Les processus client et serveur sont indépendants mais communiquent au travers de tubes nommés. S'il n'y a aucun client, le processus serveur est bloqué en attente de lecture dans le tube. Il y'a autant de tubes nommés pour les réponses que de clients. Quand un client se termine, son tube nommé est détruit. L'envoi d'un message client (une requête) consiste en une écriture dans le tube nommé du serveur. La réception d'un message client consiste en la lecture dans le tube nommé du client.