TP Tube (Partie 2)

# Description de l’application

Nous allons développer une application client/serveur en utilisant des tubes.

Le schéma de l’application est assez simple : un seul serveur accepte et traite les requêtes transmises par différents clients.

Après traitement, le serveur envoie les résultats aux clients.

Il faut permettre à plusieurs clients d’envoyer des requêtes au serveur.

Le service réalisé par le serveur sera un service d’évaluation d’expressions arithmétiques. Par exemple, si le client envoie la requête “(1+4)\*5”, le serveur évalue cette expression et retourne le résultat 25 au client concerné.

Nous allons utiliser les différents tubes vus en cours pour réaliser notre application:

* Des tubes nommés pour établir la communication entre les clients et le serveur
* Des tubes processus pour lancer l’évaluation des expressions arithmétiques par le programme bc

De plus, il faudra créer un tube par client pour que le serveur puisse retourner le résultat de l’évaluation à ce client spécifique. Cela veut dire, que la requête d’un client doit contenir un identifiant qui permet de retrouver facilement le tube nommé qui lui est associé.

Par exemple, nous pouvons utiliser la structure suivante pour représenter une requête :

#define BUFFER\_SIZE 20  
  
struct requete\_client\_serveur {  
pid\_t client\_pid; //PID du Client  
char expression[BUFFER\_SIZE - 1]; //text de l expression a evaluer.  
};

Par convention, le tube nommé qui correspond à un client de PID sera : /tmp/client\_n\_fifo.

La fifo du serveur sera par convention nommée : /tmp/serveur\_fifo

Enfin, pour évaluer les expressions arithmétiques le serveur pourra utiliser un programme externe (par exemple bc). Dans ce cas, on utilisera un tube processus pour établir la communication avec ce programme externe.

## Travail à faire

* Donnez le code du client et du serveur pour cette application