

Programmation Concurrente

Queues

Exercice 1

Créez deux programme différents (2 *main()*), l'un jouant le rôle de serveur, l'autre de client. Leurs rôles sont les suivants :

- Le client prend une chaîne de caractère en paramètre et la transmet par une queue au serveur
- Le serveur affiche la chaîne de caractère reçue

Testez la transmission.

Notes :

- effectuez le link de vos programmes avec la librairie `-lrt` et `-lpthread`
- le serveur doit créer la queue, mais pas le client
- lorsque la queue est créée, celle-ci apparaît sous forme d'un fichier virtuel ici : `/dev/mqueue/<nom de la queue>`
- lorsque `mq_close()` est exécuté, cela ne détruit pas le fichier. Pour le faire, le serveur doit appeler `mq_unlink()` après `mq_close()` pour cela.

Exercice 2

Créez un seul programme avec 2 tâches : l'une jouant le rôle du serveur, l'autre du client. Limitez le nombre d'éléments maximum de la queue à 7.

- La tâche client envoie 15 messages d'affilée au serveur (texte). Les messages doivent contenir un itérateur (1,2,3...).
- Le serveur affiche les messages reçu et attend 200 ms entre chaque affichage.

Vérifiez le bon fonctionnement de la transmission et observez l'effet de la limitation du nombre d'éléments de la queue.

Note : lorsque la queue a été créée, son nombre d'éléments reste fixe. N'oubliez pas d'appeler `mq_unlink()` à la fin du serveur