

## Exercice 1 (mkfifo, open, read, close)

Écrire une commande `serveur` permettant de créer un tube nommé dont le nom lui est passé en argument qui attend des connexions entrantes sur celui-ci lorsqu'il n'y a pas de client et d'y extraire les chaînes de caractères lues pour les afficher à l'écran. Le serveur ne doit a-priori jamais terminer. Ex. : `serveur /tmp/tube`

## Exercice 2 (open, write, close)

Écrire une commande `client` qui se connecte sur le tube nommé du serveur précédent passé en argument et y envoie les chaînes de caractères qui lui sont passées comme autres arguments. Ex.: `client /tmp/tube bonjour "au revoir"`.

Tester aussi en ralentissant les clients ou le serveur (`sleep`).

## Exercice 3

Pour établir une connexion bidirectionnelle il est nécessaire que le client puisse envoyer au serveur un tube sur lequel celui-ci renverra sa réponse et le client la lira. Écrire une commande `serveur_echo` qui permet d'attendre les connexions entrantes sur un tube dont le nom lui est passé en argument, d'y lire deux chaînes de caractères, le premier étant le nom du tube où renvoyer la chaîne correspondant au second argument. Ex. : `serveur_echo /tmp/tube`

## Exercice 4

Écrire une commande `client_echo` permettant d'envoyer des chaînes de caractères au serveur précédent de sorte qu'il les reçoive en retour sur le tube passé en argument. Ex. : `client_echo /tmp/tube /tmp/tube-reception bonjour "au revoir"`.

## Exercice 5

Modifier la commande précédente de sorte que le tube nommé de réponse soit automatiquement déterminé par le client (consulter la documentation de la fonction `tmpnam`).