

## Programmation système avancée

### TP n° 4 : Tubes nommés

#### Exercice 1 :

1. À l'aide de la commande `mkfifo` du *shell*, créez un tube nommé `tuyau`. Vérifiez à l'aide de « `ls -l` » que le tube nommé a bien été créé.
2. Tapez « `echo bonjour > tuyau` ». Que se passe-t-il ? Expliquez le comportement.
3. Pendant que la commande précédente s'exécute, ouvrez un deuxième *shell* et tapez « `cat tuyau` ». Que se passe-t-il ?
4. Quel est l'appel système qui bloque dans la question 2 ci-dessus ? Que se passe-t-il si vous essayez d'utiliser `strace` ? Utilisez la commande `tee` pour prouver votre hypothèse.
5. Répétez l'expérience précédente en inversant l'ordre des commandes (`cat` avant `echo`).

#### Exercice 2 :

Le but de cet exercice est d'écrire un démon semblable au démon `syslog` tournant sur tout système Unix.

1. Écrivez un programme nommé `log-daemon` qui crée un tube nommé « `/tmp/logger` » accessible en écriture mais pas en lecture par tout utilisateur. Votre programme lira les données qui arrivent sur la FIFO, et les affichera sur sa sortie standard (sans jamais terminer). Que doit faire votre programme lorsqu'il reçoit une indication de fin de fichier (`read` qui retourne 0) ? Que doit-il faire en cas d'erreur de lecture ? Testez votre programme à l'aide de la commande `echo`, et testez notamment qu'il peut recevoir plusieurs messages sans s'arrêter.
2. Écrivez un programme `logger` qui formate ses paramètres de ligne de commande sur la FIFO `/tmp/logger` en les terminant par un caractère de fin de ligne. Testez votre programme.
3. Modifiez le programme `log-daemon` pour qu'il précède chaque ligne qu'il affiche de la date courante (vous pouvez vous servir des fonctions `localtime` et `strftime` avec le format « `"%F %T"` »).

#### Exercice 3 :

Terminez le dernier exercice du TP précédent.