

# TD Programmation Système

## Série 3 : Synchronisation par tubes

### Exercice 1

Le programme suivant est composée d'un processus père et d'un processus fils. Le père crée un tube et écrit dans le tube, le fils lit à partir de ce tube. Après initialisations et création du tube, création du processus fils, les codes du père et du fils sont :

```
/* Père */                                /* Fils */

wait() ;                                close(p[1]) ;
close(p[0]) ;                            while( read(p[0], &c, 1) != 0)
write(p[1], "abcdef", 6) ;                {
close(p[1]) ;                            printf("%c", toupper(c)) ;
                                        }
                                        close(p[0]) ;
```

Question :

1. Ce code présente-t-il une situation d'inter-blocage ? Indiquer ce qui se produit et expliquer pourquoi.
2. Proposez une solution

### Exercice 2 : Synchronisation Père/Fils

Ecrire le programme C sous Unix qui permet à un processus père de créer un processus Fils. Le père et le Fils doivent s'exécuter en parallèle en coordonnant leurs exécutions et en permettant l'affichage suivant :

Fils	: 2	4	6	8	10
Père	: 13	16	19	22	25
Fils	: 27	29	31	33	35
Père	: 38	41	44	47	50

.....  
.....

Père : .....

On s'arrête quand  $n > 100$

Appels Systèmes : fork(), pipe(), close()

### Indication :

- Le processus Fils récupère une valeur au début égale à 0 lui ajoute successivement 2 jusqu'au premier multiple de 5, arrête son exécution et envoie la dernière valeur au processus Père.

- Le processus père récupère la valeur envoyée par le fils lui ajoute successivement 3 jusqu'au premier multiple de 5, arrête son exécution et envoie la dernière valeur au processus Fils.
- On s'arrête quand la dernière valeur affichée est  $> 100$
- Le père et le fils utilisent une communication bidirectionnelle par tube pour communiquer

Question :

1. Ecrire le programme C permettant l'implémentation des processus père et fils en utilisant les pipes
2. On désire remplacer les processus Père/Fils par deux processus sans lien de parenté proposer une solution utilisant les pipes nommés.