



TP 2 : Communication inter-processus sous UNIX

Tubes de communication

Tubes anonymes :

Les tubes anonymes sont utilisés pour des communications unidirectionnelles entre un père et ses fils. Un tube de communication permet de mémoriser des informations. Il est caractérisé par deux descripteurs en lecture et en écriture.

Syntaxe

unistd.h

```
int pipe(int p[2]) /* Création
```

Cet appel crée un tube anonyme et renvoie son descripteur de lecture dans p[0] et son descripteur en écriture dans p[1].

```
read(desc, message_lu, taille_max_message) /* Lecture du tube
```

```
write(desc_écriture, message_écrit, taille_max_message) /* Ecriture du tube
```

```
close(desc) /* Fermeture du descripteur en lecture ou en écriture du tube
```

Exercice 1 :

Ecrire un code C qui permet de créer un processus père qui à son tour crée un tube anonyme puis un fils. Le processus fils laisse un message pour son père dans le tube, puis le père récupère ce message du tube et l'affiche sur écran. Lors de la fin d'exécution, le père et le fils fermeront les descripteurs de lecture et écriture.

Tubes nommés

Les tubes nommés sont également des tubes de communication entre des processus indépendants mais plus intéressants en termes de propriétés (détails seront vus au cours).

Syntaxe

sys/types.h

sys/stat.h

```
int mkfifo (chemin_d_acces, droits d_acces)
```

Crée un tube nommé dans chemin d'accès (extension .fifo) avec les droits d'accès précisés pour le propriétaire, le groupe et tous les utilisateurs. Une fois le tube créé, il peut être utilisé pour une communication entre deux processus indépendants, l'un un writer et l'autre un reader.

```
fcntl.h   /* O_RDONLY, O_WRONLY
```

```
open(chemin_d_acces_tube, mode_d_ouverture)
```

mode_d_acces= O_RDONLY pour l'ouverture en lecture seule.

mode_d_acces= O_WRONLY pour l'ouverture en écriture seule.

```
close(desc)   /* fermeture du tube nommé en lecture ou en écriture
```

Exercice 2

Ecrire deux processus indépendants writer.c et reader.c. Le processus writer crée un tube nommé «mypipe.fifo» dans le répertoire courant avec les droits d'accès 0660 puis laisse un message pour le processus reader. Le processus reader lit le message et l'affiche sur écran. Lors de la fin d'exécution, les deux processus ferment les descripteurs en lecture et écriture.