SYSTEMES D'EXPLOITATION - TRAVAUX PRATIQUES SEANCE 5

EXERCICE 1 – Les redirections d'entrées/sorties

On souhaite écrire une commande créant **N** processus. Chacun de ces processus génère et affiche un nombre tiré aléatoirement.

Voici un exemple d'exécution de cette commande avec **6** processus. Cette commande crée **6** processus. Le nombre de processus est donné en ligne de commandes. Chacun des processus crées est identifié par un **pid** et un **numéro** d'ordre de création.

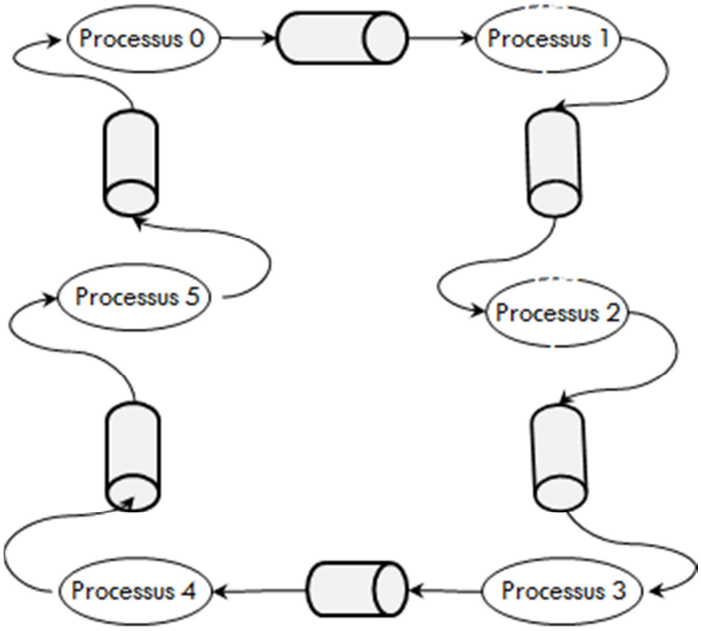
```
$ ./exo3 6
processus pid 4232 numéro  2 valeur = 1430
processus pid 4233 numéro  3 valeur = 485
processus pid 4234 numéro  4 valeur = 220
processus pid 4235 numéro  5 valeur = 2602
processus pid 4230 numéro  0 valeur = 2178
processus pid 4231 numéro  1 valeur = 1723
$
```

L'objectif est de déterminer le processus qui a généré le plus **grand** nombre. Pour cela, les **N** processus sont organisés en *anneau* au travers de leur entrée et de leur sortie standard en utilisant les **tubes anonymes**. Le processus **0** va être connecté au processus **1** par un premier tube, le processus **1** est connecté au processus **2** par un deuxième tube, le processus **2** connecté au processus **3** et ainsi de suite... Le dernier processus de l'anneau doit être connecté au processus **0**.

Lorsque les processus et les tubes sont créés, le processus **0** envoie sa valeur générée vers le processus **1**. Le processus **1** compare la valeur reçue à sa valeur et envoie au processus **2** la valeur la plus grande et chaque processus de l'anneau va appliquer le même traitement.

A la fin, le processus 0 doit recevoir sur son tube la plus grande valeur.

```
$ ./exo3 6
processus pid 4232 numéro  2 valeur = 1430
processus pid 4233 numéro  3 valeur = 485
processus pid 4234 numéro  4 valeur = 220
processus pid 4235 numéro  5 valeur = 2602
processus pid 4230 numéro  0 valeur = 2178
processus pid 4231 numéro  1 valeur = 1723
Le plus grand nombre = 2602 - pid = 4235 -  Numéro 5
$
```



Sur cet exemple, c'est le processus **5** qui a généré le plus grand nombre. Le processus **0** affiche que le gagnant est le processus numéro **5**.

EXERCICE 2 – exercice introduction aux Tubes nommés

La fonction **mkfifo()** permet de créer un tube nommé, c'est à dire un tube de communication analogue à celui créé par l'appel système **pipe()**, mais caractérisé par un chemin d'accès (comme un fichier). Commenter les fonctions suivantes, et en préciser les effets :

```
#include <sys/fcntl.h>
#include <sys/stat.h>
#include <stdio.h>
#define TAILLEMAX 50

int creer_tube() {
  return mkfifo("TubeNomme", S_IRWXU| S_IRWXG|S_IRWXO);
}

void detruire_tube() {
  unlink("TubeNomme");
}
```

Ecrire et tester un programme **lecteur.c** et un programme **ecrivain.c** pour tester les fonctions **lecture()** et **ecriture()** (on lancera simultanément plusieurs lecteurs et plusieurs écrivains).

```
void lecture() {
  int tube, longueur;
  char message[50];
  tube=open("TubeNomme",O_RDONLY);
  longueur=read(tube,message,30);
  message[longueur]='\0';
  printf("%d a lu le
message\n\t\t%s\n",getpid(),message);
  close(tube);
}

void ecriture() {
  int tube;
  char message[50];
  sprintf(message,"processus %d ",getpid());
  tube=open("TubeNomme",O_WRONLY);
  write(tube,message,strlen(message));
  close(tube);
}
```

EXERCICE 3 – Programmation d'une application Client/Serveur utilisant des tubes nommés.

La structure de l'application est assez simple : un seul serveur reçoit et traite les requêtes envoyées par différents clients. Après traitement, le serveur transmet les résultats aux clients. Il faut permettre à plusieurs clients d'envoyer des requêtes au serveur. Le service réalisé par le serveur est d´évaluer des expressions arithmétiques simples de type **a + b** (ou **a - b**). Par exemple, si le client envoie la requête **4+5**, le serveur évalue cette expression et retourne le résultat **9** au client concerné. Vous pouvez utiliser les pipes nommés pour réaliser cette application :

- Un tube pour établir la communication avec le serveur (sens client vers le serveur).
- Un tube par client pour que le serveur puisse retourner le résultat de l'évaluation. Par convention, le tube nommé qui correspond à un processus client de **PID** égal à **N** sera : **Client_N_pipe**.

La requête d'un client doit contenir un identifiant qui permet de retrouver facilement le tube nommé qui lui est associé. Par exemple, on peut utiliser la structure suivante pour représenter une requête :

```
struct requete_client_serveur {
    int clientPid;   //PID du Client
    char expression[20];
   //la chaine de caractère représentant l'expression à évaluer.
} ;
```