TD système n°2 : Communication par tubes et IPC System V

Exercice 1 : tubes ordinaires

Ecrire un programme C qui permet à un processus père de créer 2 processus fils. Le premier processus fils transmet au processus père la chaîne « Je suis le premier fils » en utilisant un tube ordinaire. Le second processus fils transmet au processus père le message « Je suis le second fils » en utilisant un tube ordinaire. Le processus père reçoit les 2 chaînes qui lui sont envoyées par ses fils et les affiche à l'écran.

Exercice 2 : ensembles de sémaphores

Soit le code C suivant :

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/ipc.h>
#include<sys/sem.h>

int main(int argc, char *argv[])
{
   key_t cle;
   int semid;
   int arg[1]={1};


if ((cle = ftok(argv[1], 5)) = = -1) {
   printf("Erreur de ftok\n");
   exit(10);
}

if ((semid = semget(cle, 1, IPC_CREAT|0666)) = = -1) {
   printf("Erreur de semget\n");
   exit(11);
}

printf("Sémaphore %d de cle %x cree \n", semid, cle);

if (semctl(semid, 0, SETVAL, arg) = = -1) {
   printf("Erreur de semctl\n");
   exit(11);
}
}
```

1. Décrire le fonctionnement de ce code.
2. Ce code est utilisé pour simuler le comportement d'un guichet unique. Écrire le code d'un client qui souhaiterait accéder à ce guichet, sachant que le code du guichet est exécuté avant celui du client.

Exercice 3 : segments de mémoire partagée

Un processus fils « écrit » toutes les 2 secondes un numéro tiré au hasard entre 0 et 49 dans une zone de mémoire partagée créée par le père avant la création du fils.

Le père additionne ces numéros. Dès que le total dépasse 1000, il envoie le signal `SIGUSR1` au fils.

À la réception du signal `SIGUSR1` le fils s'arrête. Après la fin du fils, le père détruit le segment de mémoire partagée et s'arrête aussi.

Écrire le code du père et du fils.

Exercice 4 : client/serveur et file de messages

Un programme serveur créé une file de messages et se met ensuite en attente d'éventuelles communications demandées par des processus clients. Ce serveur reçoit des messages de type 1 dans lesquels les processus clients indiquent leur pid. Le serveur répond aux clients en leur renvoyant un message dont le type est le pid du destinataire. Les clients n'acceptent que les messages portant leur propre numéro de pid.

En pratique, on fait tourner ce serveur de messages, puis on lance les clients à partir d'un autre terminal.

Écrire le code du serveur et d'un client.