

Programmation système

TD n° 1

Multi-processing

1 Exécution concurrente des processus père et fils

Écrire un programme qui crée un fils :

- Le père affiche : “Je suis le père mon PID est ... et le PID de mon fils est ...”
- Le fils affiche : “Je suis le fils mon PID est ... et le PID de mon père est ...”

Rajouter l’appel à la primitive `sleep` pour voir les exécutions possibles :

1. le fils se termine avant le père,
2. le père se termine avant le fils.

2 Cas particulier des descripteurs de fichiers

Écrire un programme qui ouvre un fichier nommé `toto`, en lecture et écriture, dont le contenu est la suite 123456789. Le programme fork ensuite ; le fils écrit `ab` dans le fichier ensuite il s’endort et il lit 2 caractères ; le père s’endort, lit 2 caractères et écrit `AB`.

3 Déroutement d’un signal

Écrire un programme qui crée un fils. Le père dérouté les signaux `SIGTERM` et `SIGKILL`, puis s’endort. Le fils lui envoie successivement ces deux signaux. Que constatez-vous ?

4 Synchronisation par `SIGSTOP` et `SIGCONT`

Écrire un programme qui crée un fils. Le père doit effectuer deux tâches `p1` et `p2`. Le fils doit effectuer deux tâches `f1` et `f2`. On veut synchroniser la réalisation de ces tâches dans cette ordre : `f1`, `p1`, `f2`, `p2` en utilisant les signaux `SIGSTOP` et `SIGCONT`.