

IN405 – Feuille de TD #5

Processus

Exercice 5.1 - Création simple de processus

En utilisant la fonction `fork`, écrivez les programmes correspondant aux comportements suivants pour un processus père et son processus fils :

- 1- Affichage de 'Hello World!' pour les deux processus.
- 2- Affichage de 'Mon PID est ... et celui de mon père/fils est ... !'.
- 3- Le processus fils choisit aléatoirement un nombre entre 1 et 50, l'affiche, puis le communique à son père qui l'affiche à son tour.

Exercice 5.2 - `sleep()` & `wait()`

Écrivez le programme correspondant à l'énoncé suivant :

- 4- Le processus père crée 10 processus fils et attend qu'ils se terminent. Chaque fils attend un nombre de secondes choisi aléatoirement entre 1 et 10, affiche son PID puis se termine. Le processus père affiche à chaque terminaison, le PID du processus fils qui a terminé son exécution.

Exercice 5.3 - Création multiple de processus

Écrivez les programmes correspondants aux énoncés suivants, avec m et n , deux entiers donnés au lancement du programme. Pour chacun d'entre eux, calculez, à l'aide du programme, le nombre total de processus créés.

- 5- Le processus père crée m fois n processus fils.
- 6- Le processus père crée m processus fils, puis chaque processus fils crée n processus petit-fils.

7- Le processus père crée n processus fils, puis chaque processus fils crée n processus petit-fils, puis chaque processus petit-fils etc., ceci m fois.

Exercice 5.4 - 2, 3, 5, 7, etc.

Soit l'algorithme donné en 5.1, répondez aux énoncés suivants.

Listing 5.1: PREMIER algorithme

```
1  entree :  $n$ 
2  sortie :  $r$ 
3
4   $i \leftarrow 2$ 
5   $r \leftarrow \text{vrai}$ 
6  tant que  $i \leq \sqrt{n}$ 
7    si  $n \bmod i \equiv 0$ 
8       $r \leftarrow \text{faux}$ 
9     $i \leftarrow i + 1$ 
```

8- Compréhension – Que fait cet algorithme ?

9- Théorie – Déterminez la complexité en nombre d'itérations de l'algorithme pour une exécution sur le nombre n . Faites en de même pour une exécution sur l'ensemble des nombres 1 à 10, puis 1 à 100. Comparez les exécutions sur l'ensemble des nombres 1 à 100, 101 à 200, 201 à 300, etc.

10- Pratique – Implémentez un programme qui exécute l'algorithme sur un ensemble n donné de nombres. Parallélisez ce programme de sorte que P_1 traite les n/m premiers nombres, P_2 traite les n/m suivants, ..., P_m traite les n/m derniers. Affichez le temps utilisé par chaque processus.