Gestion des processus sous UNIX

Gestion de la priorité

Gestion de la priorité

- **Lors du TP précédent où nous avions créé deux processus, père et fils, certains ont pu constater que l'ordre d'exécution des instructions de chacun n'était pas déterminé à l'avance et que celui-ci était quelque peu aléatoire.**
- **Nous allons tenter de mieux comprendre ce phénomène et même de mieux le maîtriser.**

- Nous avons vu que l’ordre d’exécution des processus est déterminé par un ordonnanceur qui choisi le processus à exécuter par le processeur central en fonction de sa priorité et de son état (s’il est prêt ou en attente).
- Donner la commande shell qui permet de connaître la priorité des processus (PRI) : _____
- Donner la fourchette des valeurs relevées : de ____ à _____

Modification de la priorité

- A partir du programme précédent créant un processus fils et père, modifiez leur exécution afin qu'ils affichent chacun un message de manière cyclique sur l'écran. Le père affichant à gauche et le fils à droite.
- On désire à présent influer sur la priorité de chaque processus.

- Pour cela on utilise la primitive nice(), dont la syntaxe est la suivante :

```
#include <unistd.h>

int nice (int incr);
```

- **Rôle:**

  Faire varier la priorité d'exécution du processus appelant en augmentant/diminuant la priorité de la valeur **incr** (incrément).

- **Argument:**

  incr est la valeur à ajouter/retrancher à la valeur de la priorité courante.

- **Description:**

  Pour un processus superviseur (root), **incr** est une valeur quelconque positive ou négative.

Travail demandé :

- Faire un programme réalisant deux processus à l'aide de fork()
- Diminuer la priorité de l'un des deux processus à l'aide de nice()
- Afficher les valeurs de PRI et NI à l'aide de system(« ps –l ») avant et après l'exécution de nice()
- Quel est le lien entre la valeur passée en paramètre et
  - la valeur affichée ?
  - la priorité du processus (PRI) ?
  - le temps d'exécution ?
- **Conclusion ?**

- Réalisez un programme permettant d'observer l’influence de nice sur la vitesse d'exécution des processus et sur l’affichage. Résultat attendu :

EXECUTION PERE

EXECUTION PERE

EXECUTION PERE

EXECUTION PERE

EXECUTION FILS

EXECUTION FILS

EXECUTION FILS

EXECUTION FILS

- On réalisera pour cela une boucle de 100000 affichage, mais la fonction nice() ne doit être appelée qu’une fois avant d’entrer dans la boucle

- Observer l’influence sur la vitesse d’exécution de chaque processus