Sujet du TP n° 4

Préambule

Placez-vous dans le répertoire **PSE/TP4**.

Les processus (transparents 72 à 93)

Pratique 1

- ouvrez un **terminal** et lancer **gedit**
- dans un autre **terminal** lancez la commande ps avec l'option l : **ps l**
- grâce aux colonnes **PID** et **PPID** visualiser la généalogie des processus
- pour visualiser le père des shell des deux terminaux exécutez **ps xl** (affiche aussi les processus non contrôlés par un terminal)

Exercice 1

But : appréhender de manière simple et progressive la programmation des mécanismes fork/exec/wait. Tester à chaque étape.

A. Père et fils dans un même programme.

1. On veut créér un programme dans lequel un processus père crèe 2 processus fils. Pourquoi est-il incorrect d'écrire le code ci-dessous ?

```
...
fork();
fork();
...
```

2. Ecrire la version correcte du programme où un père crèe 2 fils, chaque fils affiche un message.
3. Mettre une temporisation de 10 s (fonction sleep(10)) à la fin du père et des fils, pour avoir le temps de lancer dans un autre terminal la commande ps l et visualiser la généalogie des processus.

B. Père et fils dans des programmes différents.

1. Mettre les instructions des fils dans des programmes indépendants F1 et F2. Visualiser les processus par la commande ps l.
2. Passer à F1 un argument quelconque qu'il affiche.
3. Faire afficher par le père son pid et celui de ses fils, et par chaque fils son pid et celui de son père.
4. Faire en sorte que le père attende la fin de chaque fils et indique à chaque fois lequel s'est terminé; on peut enlever la temporisation de 10 s à la fin du père.
5. Faire en sorte que chaque fils se termine avec un code d'exit et que le père récupère ces codes et les affiche.

Exercice 2

Ecrire un programme qui met en oeuvre l'algorithme ci-dessous où il lance gedit dans un processus séparé puis réalise un dialogue opérateur de multiplication de deux nombres.
L'exécution de gedit et le dialogue de calcul n'ont aucun lien fonctionnel entre eux. Ils servent juste à avoir deux processus en parallèle pouvant être contrôlés par l'opérateur.
Indication : pour connaitre l'emplacement de l'exécutable gedit utiliser la commande which, **which gedit**.

```
saisir un nom de fichier
créer un processus qui lance gedit sur ce fichier
réaliser le dialogue : saisir deux nombres et afficher leur produit
si edition du fichier non terminee (waitpid, voir transparent 91)
  demander à l'opérateur s'il veut attendre
  si oui
      attendre la fin de l'édition
  finsi
finsi
```

Pratique 2

Une fois le programme de l'exercice 2 testé, faire les actions ci-dessous illustrant certains états d'un processus fils.

- Se placer dans le cas où l'opérateur choisit de ne pas attendre la fin de l'édition, puis exécutez la commande ps l.
Le fils (gedit) existe toujours car la fin d'un processus père n'entraine pas la fin de ses fils.
Mais il est dans l'état dit orphelin qui le rattache au processus initial du système de PID 1 ou un fils de celui-ci.

- Arrêter l'éditeur alors que le processus père est en attente de saisie des nombres et dans un autre terminal exécutez la commande ps l.
Le processus fils (gedit) apparaît toujours dans la liste, mais dans un état "defunct", dit aussi "zombie".
Dans cet état le fils est terminé mais son père, qui n'est pas terminé, n'a pas (encore) attendu la fin du fils.
Une fois que le père a attendu la fin du fils ou se termine, le fils disparaît de la liste des processus.