

# Programmation Système

## TP n° 1 : création de processus

### 25-26 janvier 2016

#### Exercice 1 : cacophonie

Écrire un programme qui crée deux processus : un père qui affiche à l'aide de l'appel système `write()` la chaîne "tut ", et un fils qui affiche la chaîne "pouet "; vous n'omettez pas d'attendre la mort du fils à l'aide de `wait()`. Quel est l'ordre des chaînes affichées ? Un autre résultat est-il possible ?

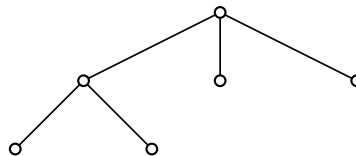
Modifier ce programme pour que les deux processus écrivent "tut " ou "pouet " de façon ininterrompue (dans une boucle). Interpréter l'affichage. On finira par un bon `ctrl-c` !

#### Exercice 2 : `fork()` dans une boucle

1. Écrire un programme qui crée treize processus fils dont chacun affiche son *pid*. Vérifiez que votre programme affiche bien treize lignes.
2. Modifier le programme pour que le père attende que les fils soient terminés, et annonce les décès au fur et à mesure.
3. Modifier le programme pour que chaque fils attende un temps aléatoire<sup>1</sup>, tiré entre 2 et 20 secondes, avant de réaliser son affichage et de mourir. Que se passe-t-il ? Pourquoi ? Si tous les fils tirent le même nombre « aléatoire », corrigez le bug...

#### Exercice 3 :

Écrire un programme qui crée l'arbre de processus ci-dessous.



Vérifier à l'aide de la commande « `pstree` ».

#### Exercice 4 :

1. Écrire un programme qui génère un tableau de 10 000 entiers tirés au hasard entre 0 et 1000 puis affiche tous les indices des cases de ce tableau dont la valeur est 42, et enfin le nombre total de « 42 » trouvés.
2. Modifier votre programme pour qu'il utilise deux processus, le père qui parcourt la première moitié du tableau, et le fils qui parcourt la seconde.
3. Même question avec  $2^k$  processus, où  $k$  est donné sur la ligne de commande.

#### Exercice 5 :

Écrire un programme qui simule la commande « `find . -name fic` » de manière parallèle, *i.e.* en déléguant la recherche dans chaque sous-répertoire à un fils créé pour l'occasion.

1. vous pouvez utiliser la fonction `random()`.