

TP UNIX n°1

Pour compiler vos programmes utiliser `gcc xxx.c -o xxx`

N'hésitez pas à utiliser la commande **man** pour obtenir de l'aide sur les fonctions et commandes

ATTENTION pour tous ces exercices n'oubliez pas de mettre un '\n' à la fin de vos printf pour être sûrs qu'ils soient faits immédiatement.

1°) *Fonctions utiles : fork , getpid , getppid , exit, sleep, printf*

Ecrire un programme qui :

- affiche : "je suis le processus numéro x" où x est son numéro de processus
- puis récupère dans une variable la valeur retournée par la fonction fork
- puis affiche : "fork m'a renvoyé la valeur : x" où x est le contenu de cette variable
- puis affiche : "je suis le processus numéro x et mon père est le processus numéro y" où x est son numéro de processus et y celui de son père

Combien d'affichages a-t-on ?

A quoi correspondent-ils ?

2°) Ecrire un processus qui crée un processus fils par fork.

Le processus père affiche les 10 premiers nombres impairs par une boucle for

Le processus fils affiche les 10 premiers nombres pairs par une boucle for

a) exécuter ce programme et voir comment se font les affichages.

b) Ajouter maintenant une attente (`sleep(1)`) dans la boucle d'affichage du processus père, que se passe-t-il ?

c) Enlever l'attente dans la boucle du père et la mettre dans la boucle du fils. Lancer ce programme. Que se passe-t-il ? Pourquoi ?

Lancer à nouveau ce programme et pendant qu'il affiche des nombres pairs exécuter la commande `ps -l`. On voit que le processus fils continue à fonctionner mais qui est son père ?

Quel est ce processus ?

Fonctions utiles : fork , execlp , execl , exit, sleep, printf, wait

3°) Ecrire un processus qui crée un processus fils par fork

Le processus fils exécute la commande unix 'ps' à l'aide de la fonction `execlp`

Le processus père attend la fin de son fils et affiche la valeur que lui a retournée son fils.

4°) Remplacer dans le programme du 3°) l'appel à `execlp` par un appel à `execl` que se passe-t-il pourquoi ?

5°) Modifier le programme du 3°) pour que la commande exécutée par le fils ainsi que les paramètres de cette commande soient passés en paramètre au programme.

Tester votre programme en tapant : `./votre_programme ls -li`

et vérifier que le processus fils tient bien compte de l'option `-li` de la commande `ls` qui affiche le répertoire courant en donnant, pour chaque fichier, son numéro d'i-node.