TD3 Processus : fork()

Rappelle : la fonction fork() permet de créer un processus fils.

int pid=fork();

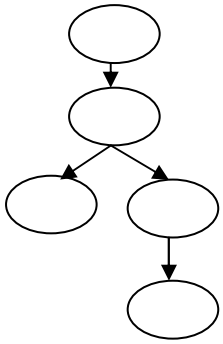
Si (pid == 0) alors vous êtes le fils sinon vous êtes le père et le pid de votre fils est stocké dans la variable pid.

Exercice 1 :

Donner un programme *c* qui crée 3 processus fils. Chaque fils affiche son PID et celui de son père. Les 3 fils doivent être frères. Une fois les trois processus fils exécutés, le père indique que tout va bien.

Exercice 2 :

Donner un programme *c* qui crée les processus vérifiant l'arbre des processus suivant :
(On considère que la racine est le terminal).



Chaque processus doit afficher son PID et celui de son père.

Exercice 3 :

Donner un programme *c* qui crée 2 processus fils. Chaque processus fils affiche son pid toutes les secondes.

Exercice 4 :

Donner un programme *c* qui crée 2 processus fils. Les deux processus fils vous demandent régulièrement (toutes les 1 à 2) un entier, puis quand vous le saisissez, l'affiche en indiquant en même temps son PID.

Exercice 5 :

Créer un programme *c* qui affiche votre nom et prénom et un second programme qui affiche une phrase.

Exercice 6 :

Donner un programme *c* qui crée 2 processus fils. Le premier exécute le premier programme de l'exercice 5 et le second le deuxième programme de l'exercice 5.

Il faut la fonction execve(commande, argument, environnement) pour lancerun programme dans un processus (voir le manuel de cette fonction !).

Exercice 7 :

Donner un programme *c* qui demande à l'utilisateur un programme à exécuter, l'exécute puis en demande un à nouveau et ainsi de suite.

Exercice 8 :

Donner un programme *c* qui crée 3 processus fils. Chaque processus fils demande toutes les secondes un entier et l'ajoute à ceux déjà saisis, puis l'affiche. Chaque processus fils demande 10 entiers.

Attention : chaque processus, quand il parle sur le terminal, indique son numéro. Le premier fils s'appellera 1, le deuxième 2 et le troisième 3. (Cela ne correspond pas à leurs PID).

Exercice 9 :

Donner un programme *c* qui exécute tous les programmes donnés en argument.

Rappel : int main (int argc, char *argv[]) ; argc le nombre d'arguments ; *argv[] le tableau de chaines de caractères.