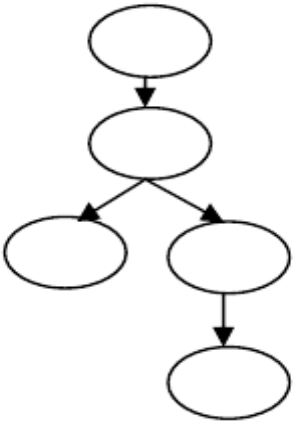
Système TD 3

EXERCICE 1

Écrire 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 3 processus fils exécutés, le père indique que tout va bien.

EXERCICE 2

Écrire un programme C qui crée les processus vérifiant l'arbre des processus suivant :



EXERCICE 3

Écrire un programme C qui crée 2 processus fils. Chaque processus fils affiche son PID toutes les secondes.

EXERCICE 4

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

EXERCICE 5

A. Écrire un programme C qui affiche votre prénom et nom.
B. Écrire un programme C qui affiche une phrase.

EXERCICE 6

Écrire un programme C qui crée 2 processus fils.
Le premier exécute le programme 5.A, et le second exécute le programme 5.B.

Aide : il faut utiliser la fonction *execv(commande, argument)* pour lancer un programme.

EXERCICE 7

Écrire un programme C qui demande à l'utilisateur un programme à exécuter, puis l'exécute dans un processus fils, puis en demande un à nouveau, et ainsi de suite.

EXERCICE 8

Écrire 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 affiche le total. Chaque processus fils demande 10 entiers.

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

EXERCICE 9

Écrire un programme C qui exécute (dans un processus fils) tous les programmes passés en argument (*argv*).

Rappel :
argc : le nombre d'arguments
argv : les arguments, sous forme de tableau de chaînes de caractères