

TP 6 : Création de processus**Exercice 1 :**

Écris un programme en C qui affiche le numéro d'identification (pid) du processus associé et celui de son père, en utilisant les primitives getpid() et getppid().

Exercice 2 :

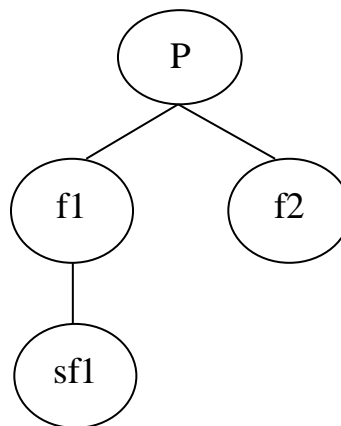
Écris un programme en C qui crée deux processus avec la primitive fork().

Le processus père affiche « Je suis le père » avec son pid, le pid de son père et le PID de son fils.

Le processus fils affiche « Je suis le fils » avec son pid et le pid de son père.

Exercice 3 :

Écrire un programme qui engendre l'arbre généalogique suivant :



1) En plus, on veut que :

- Le processus père (P) affiche « Je suis le père ».
- Le 1^{er} processus fils (f1) du (P) affiche « Je suis le 1er fils ».
- Le 2^{eme} processus fils (f2) du (P) affiche « Je suis le 2eme fils ».
- Le processus sous fils (sf1) du (f1) affiche « Je suis le sous fils ».
- Les quatre processus affiche « Traitement terminé » à la fin de leur exécution.

2) Modifier le code pour ajouter un 3^{eme} processus fils (f3) au processus père (P).

Le 3^{eme} processus fils (f3) affiche « Je suis le 3eme fils » et « Traitement terminé » à la fin de son exécution.

Exercice 4 :

Dessiner l'arbre généalogique des processus engendrés par le programme suivant.

```
#include <unistd.h>
int main()
{
    if(fork()==0)
        if(fork()!=0)
            fork();
    fork();
    return 0;
}
```

Exercice 5 :

Considérez le programme C ci-après :

```
#include <stdio.h>
int main(int arg, char **argv)
{
    int variable = 20;
    if(fork()==0)
        variable = variable + 10;
    else
        variable = variable + 30;
    printf("variable = %d\n", variable);
    return 0;
}
```

Décrivez le déroulement de ce programme. Quelles en sont les sorties (affichages) possibles ?

Exercice 6 :

Considérez le programme C suivant :

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int a = 10;
void main()
{
    pid_t pid;
    switch (pid = fork())
    {
        case -1: printf("le fork a échoué !"); break;
        case 0 : printf("Ici processus fils!, Mon PID %d\n",getpid());
                  a+=100;
                  break;
        default: printf("Ici processus père!, Mon PID %d\n",getpid());
                  a-=100;
                  break;
    }
    printf("Fin du processus %d avec a = %d\n",getpid(),a);
}
```

Décrivez le déroulement de ce programme, ainsi que les affichages générés.