

Exercice 1 :

Combien de processus le programme suivant crée-t-il ?

```
#include<stdio.h>
#include<unistd.h>
#include<stdlib.h>
int main() {
    fork();
    fork();
    fork();
return EXIT_SUCCESS;
}
```

Exercice 2 :

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

```
# include <unistd.h>
# include <stdio.h>
int main() {
    pid_t pid; int i;
    for (i=0; i<3;i++) {
        pid = fork();
        if (pid < 0){
            printf ("le fork ( ) a échoué \n" );
        }
        else if (pid == 0){
            printf(" je suis le processus : %d, mon père est : %d\n", getpid(), getppid() );
        }
        else{
            printf("je suis le processus : %d, mon père est : %d\n", getpid(), getppid() );
        }
    } return 0 ;
}
```

Exercice 3 :

Écrire un programme en C qui lance 5 fils et attend la fin de leur exécution pour se terminer.

Exercice 4 :

Écrire un algorithme puis réaliser un découpage en tâches t_1, \dots, t_n de l'expression suivante:

$$y := ((a+b) / (c - d) + (e*f)) + (a+b) * (c-d)$$

En vous servant de la définition de la condition de Bernstein, étudier la possibilité de paralléliser cette expression