

## Exercice 2

1. Ecrivons un programme qui illustre la création de 5 processus fils qui se contentent d'afficher leur numéro (compris de 0 à 4), ainsi que leur PID.

```
najma@najma:~/Desktop/tpse_aminata_yade$ gcc prog5.c -o prog5
najma@najma:~/Desktop/tpse_aminata_yade$ ./prog5
Je suis le processus fils 0 avec PID 4500
Je suis le processus fils 1 avec PID 4503
Je suis le processus fils 2 avec PID 4516
Je suis le processus fils 3 avec PID 4539
Je suis le processus fils 4 avec PID 4549
Je suis le processus parent avec PID 4499
najma@najma:~/Desktop/tpse_aminata_yade$
```

Ce programme utilise une boucle `for` pour créer 5 processus fils. Chaque processus fils affiche son numéro (de 0 à 4) ainsi que son PID. Le processus parent affiche également son PID après la création des fils. L'utilisation de `sleep(1)` dans le processus parent aide à séparer les messages dans la sortie pour une meilleure lisibilité.

2. Écrivons maintenant un programme qui permet la création de N processus fils, avec I passe en paramètre

On a remplacé N par la valeur souhaitée. Ce programme crée N processus fils, et chaque fils affiche son numéro i, son PID et son PPID avant de se terminer. Le père attend la fin de chaque fils avant de créer le suivant.

```
najma@najma:~/Desktop/tpse_aminata_yade$ gcc prog6.c -o prog6
najma@najma:~/Desktop/tpse_aminata_yade$ ./prog6 7
Fils 0 - PID: 5202, PPID: 5201
Fils 3 - PID: 5205, PPID: 5201
Fils 2 - PID: 5204, PPID: 5201
Fils 4 - PID: 5206, PPID: 5201
Fils 1 - PID: 5203, PPID: 5201
Fils 5 - PID: 5207, PPID: 5201
Fils 6 - PID: 5208, PPID: 5201
Père - PID: 5201
PID des fils morts : 5202 5203 5204 5205 5206 5207 5208
najma@najma:~/Desktop/tpse_aminata_yade$
```

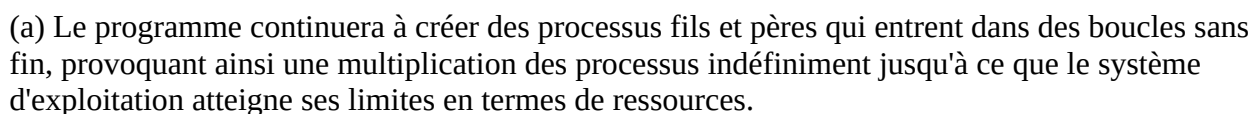
## Exercice 3 :

Ecrivons un programme illustrant l'appel système `wait` : Un processus fils est créé et le processus père attend la fin de l'exécution de son fils.

```
najma@najma:~/Desktop/tpse_aminata_yade$ gcc prog7.c -o prog7
najma@najma:~/Desktop/tpse_aminata_yade$ ./prog7
Fils 0 - PID: 6006, PPID: 6005
Le fils avec PID 6006 s'est terminé normalement. Valeur de retour : 0
najma@najma:~/Desktop/tpse_aminata_yade$
```

Ce programme crée cinq fils qui se mettent en pause pendant un temps aléatoire avant de se terminer. Le père attend la fin de chaque fils dans l'ordre.

### Exercise 5



b(i). On tape la commande `kill -9 593` . Qu'est-ce-qui se passera, que signifie cette commande?

La commande « kill - 9 593 » envoie un signal 9 ou SIGKILL pour tuer le processus essai dont le PID = 593 si nous avons les droits.

b(ii). Que pouvez-vous dire de l'affichage de la commande ps -l après exécution de la commande kill -9 593 ?

L'affichage de la commande « ps -l » après l'exécution de la commande « kill -9 593 » montre que le processus essai est bien mort mais qu'il est toujours présent dans la table des processus : il est à l'état zombie. Le processus père n'a donc pas traité le signal SIGCHLD qui lui a été envoyé.

c(i). Que signifie cette commande kill -9 604 ? Que pouvez-vous dire de l'affichage de la commande ps -l

La commande `kill -9 604` envoie un signal 9 (SIGKILL) au processus avec le PID 604, ce qui provoque une terminaison immédiate et forcée du processus. Ce signal est non bloquable et ne permet pas au processus de faire des actions de nettoyage avant de se terminer.

Lorsqu'un processus se termine, il peut devenir un processus zombie jusqu'à ce que son parent récupère son descripteur de processus. Cependant, le signal SIGKILL ne génère pas de processus zombie, car il tue le processus instantanément sans laisser de trace zombie.

La commande `ps -l` après l'exécution de `kill -9 604` montrera probablement que le processus avec le PID 604 n'est plus présent dans la liste, car il a été tué de manière immédiate et ne laisse pas de processus zombie. Cela indique que le processus a bien été terminé par le signal SIGKILL, et son descripteur de processus a été libéré sans rester en état zombie.

### **3 execution du programme modifie**

Le programme essai affiche apres execution de facon infini la phrase « je suis le fils »

#### **3a. Que signifie cette commande kill -9 628 ? Que pouvez-vous dire de l'affichage de la commande ps -l**

La commande `kill -9 628` envoie un signal 9 (SIGKILL) au processus avec le PID 628, ce qui signifie qu'elle force la terminaison immédiate et sans aucune possibilité de résistance du processus.

Lorsqu'un processus se termine, il peut devenir un processus zombie jusqu'à ce que son parent récupère son descripteur de processus. Cependant, le signal SIGKILL ne génère pas de processus zombie, car il tue le processus instantanément sans laisser de trace zombie.

Si la commande `ps -l` montre que le processus 628 a bien été tué, cela signifie que le processus père a probablement traité le signal SIGCHLD, a nettoyé le descripteur de processus du processus fils, et le processus 628 n'est plus actif.

En termes simples, `kill -9 628` force l'arrêt immédiat du processus avec le PID 628, et l'affichage de `ps -l` confirme que le processus a été effectivement terminé.

