

# TP 04 - Processus sous Linux -

## 1 Définition

Un processus est un programme en cours d'exécution.

## 2 Gestion des processus

### 2.1 Identification d'un processus

Un processus est identifié de manière unique par un numéro appelé *PID* (*Process Identifier*). La fonction *getpid* retourne le PID du processus qui l'exécute.

La fonction *getppid* nous permet de récupérer le PID du processus père qui a lancé le processus courant.

#### Syntaxe

```
#include<sys/types.h>
#include<unistd.h>

pid_t getpid();
pid_t getppid();
```

### 2.2 Création d'un nouveau processus

Un processus est créé par l'instruction *fork*. Le processus créé (*fil*) est un clone (copie conforme) du processus créateur (*père*). Le père et le fils ne se distinguent que par le résultat retourné par le *fork*. Pour le père, cette fonction renvoie le numéro du fils (-1 en cas d'erreur) et pour le fils elle renvoie 0.

#### Syntaxe

```
#include<sys/types.h>
#include<unistd.h>

pid_t fork();
```

### 2.3 Attendre d'un processus

La fonction *wait* attend et récupère la valeur passée à *exit* par le processus fils.

## Syntaxe

```
#include <sys/types.h>
#include <sys/wait.h>
```

```
pid_t wait(int *status);
```

Lorsque l'on appelle cette fonction, cette dernière bloque le processus à partir duquel elle a été appelée jusqu'à ce qu'un de ses fils se termine. Elle renvoie alors le PID de ce dernier. En cas d'erreur, la fonction renvoie la valeur -1. Le paramètre *status* correspond au code de retour du processus.

**Remarque** Il faut mettre autant d'appels de *wait* qu'il y a de fils.

## 3 Exercices

### Exercice 1

1. Écrire un programme qui crée deux processus fils appelés *fil 1* et *fil 2*. Le père doit afficher le message *je suis le père*, le fils 1 doit afficher *je suis le fil 1* et le fils 2 doit afficher *je suis le fil 2*;
2. Modifier ce programme pour que le fils 1 affiche les nombres de 1 à 50 et le fils 2 affiche les nombres de 51 à 100;
3. Lancer le programme plusieurs fois et observer l'ordre d'affichage;
4. Modifier ce programme pour que l'affichage soit dans l'ordre.

**Exercice 2** Écrire un programme qui crée  $n$  processus fils ( $n$  est passé comme paramètre au programme). Chaque processus fils doit afficher le message *je suis le fil  $i$* , où  $i$  est le numéro correspondant à chaque processus fils.