

## TD N°1 : Gestion des processus – Manipulation des signaux

### Exercice 1

- Quelle est la différence entre les appels système `wait()` et `waitpid()` ?
- Ecrire un programme C qui permet à un processus de créer 2 processus fils. Les processus fils effectuent le traitement suivant :

- affichage d'un message indiquant qu'ils sont les fils et précisant leur PID,
- temporisation puis fin.

Le processus père effectue le traitement suivant :

- affichage d'un message indiquant qu'il est le père et précisant son PID,
- attente de la mort de son premier fils crée,
- affichage d'un message indiquant le réveil du père après la mort de son fils, puis fin.

Remarque : un processus peut temporiser en exécutant l'appel

```
#include <unistd.h>
int sleep(int seconds) ;
```

### Exercice 2

Donner un exemple d'exécution du programme C suivant :

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

int main (void)
{
    int n = 100 ;

    printf(«  bonjour --> ») ;
    n *= 2 ;
    if (fork() == 0) {
        sleep(1) ;
        printf(«  dans le fils, adresse de n = %p\n » , &n) ;
        n += 10 ;
        sleep(1) ;
        printf(«  n = %d\n », n) ;
    }
    else {
        printf(«  dans le pere, adresse de n = %p\n », &n);
        n += 20 ;
        sleep(3) ;
        printf(«  n = %d\n », n) ;
    }
    exit(0) ;
}
```

### Exercice 3

Ecrire un programme C qui permet à un processus de créer dix processus fils. Chaque processus fils affiche le double de la valeur de son PID à l'écran et se termine.

### Exercice 4

1. Ecrire un programme qui permet d'ignorer le signal correspondant à la frappe des touches « CNTRL C » (signal SIGINT).
2. Ecrire un programme qui réalise un déroutement du signal correspondant à la frappe des touches « CNTRL C ». Le déroutement consiste à afficher le message « Signal reçu » en indiquant le numéro du signal.
3. Ecrire un programme qui réalise un déroutement du signal associé à la frappe des touches « CNTRL C ». Le déroutement consiste à compter le nombre de signaux reçus et à afficher le message « Signal reçu n fois » en indiquant le numéro du signal. Le 5<sup>ième</sup> « CNTRL C » termine le processus.

Pour chaque question ci-dessus :

- a. utiliser *signal()* pour installer le gestionnaire.
- b. utiliser *sigaction()* pour installer le gestionnaire.

### Exercice 5

Les signaux SIGUSR1 et SIGUSR2 sont deux des signaux standards définis par la norme POSIX. Il s'agit de signaux émis par un processus utilisateur. Ecrire un programme en C qui permet à un processus :

- de créer un processus fils,
- d'envoyer un signal SIGUSR1 à son processus fils après avoir testé son existence avec l'appel système approprié,
- d'attendre la terminaison de ce processus fils et de récupérer son « status » à la terminaison.