UNIVERSITÉ CAEN NORMANDIE

TD/TP : Les signaux

31 octobre 2018

1 Introduction

Dans ce TD/TP, nous allons voir comment sont gérés les signaux, quels sont les problèmes qu'ils apportent et comment les résoudre.

2 Exercice 1

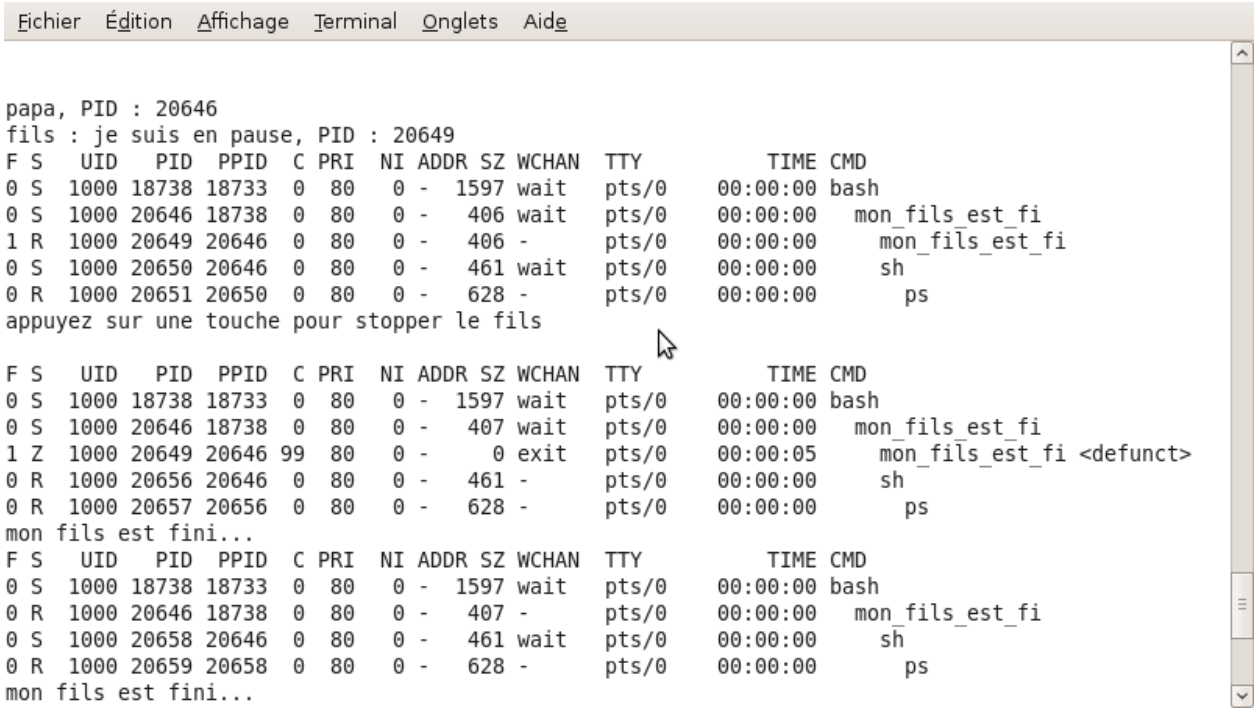
L'objectif de ce premier exercice est de comprendre certains mécanismes.

1. Soit le code suivant dont le résultat est donné ci-contre,

```
/* fichier mon_fils_est_fini.c */
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <signal.h>
#include <sys/types.h>
#include <sys/wait.h>

void mon_fils_est_fini(int sig){
  printf("mon fils est terminé... \n");
}

int main(){
  pid_t pid;
  system("clear"); // une belle console...
  if( (pid = fork())==0){ // Dans le fils
    printf("fils : je me met en pause, mon PID est: %d\n", getpid());
    for(;;); // boucle infinie
  }
  /* le père stoppe son fils*/
  printf("père, mon PID : %d\n",getpid());
  system("ps -lH");
  (void) signal(SIGCHLD, mon_fils_est_fini);
  printf("appuyez sur une touche pour stopper le fils\n");
  getchar();
  kill(pid, SIGINT); //le père interrompt le fils pid
  system("ps -lH");
  waitpid(pid, NULL, 0);
  system("ps -lH");
  return EXIT_SUCCESS;
}
```



1. Pourquoi y a t-il 2 fois le message "*mon fils est terminé...*" d'affiché ?
2. **Modifiez** le programme, pour que cela n'arrive pas.
3. **Concluez**.

3 Exercice 2

Vous utilisez la primitive **signal** pour cet exercice.

Écrivez un programme C dont le comportement est une boucle infinie et qui envoie sur STDOUT le message BONJOUR lorsqu'il reçoit le signal SIGURS1 et le message BONSOIR lorsqu'il reçoit le signal SIGURS2.

Pour SIGURS1 cela ne doit se produire qu'une seule fois par contre il doit réagir autant de fois que le signal SIGURS2 lui arrive.

Pour lui envoyer les signaux utilisez la commande :

 # kill -10 nŕpid pour SIGURS1

 # kill -12 nŕpid pour SIGURS2

4 Exercice 3

Vous utilisez la primitive **sigaction** pour cet exercice.

1. **Écrivez** un programme C permettant d'empêcher qu'un processus soit tué lorsqu'il exécute une division par zéro.

   Ce programme explique ce qui s'est passé.

   Dans un premier temps on autorisera qu'une seule fois cette erreur.
2. **Modifiez** le code afin que si l'***exception*** se reproduit, on explique à nouveau et en permanence.

5 Concluez