

## TRAVAUX PRATIQUE N° 5 (LES SIGNAUX)

### Exercice1 :

(a) Affichez le nom et les numéros des signaux.

```
janati@ubuntu: ~  
ezzouhri.A@janati_idrissi.M:~$ kill -l  
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP  
6) SIGABRT     7) SIGBUS     8) SIGFPE     9) SIGKILL     10) SIGUSR1  
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE    14) SIGALRM     15) SIGTERM  
16) SIGSTKFLT  17) SIGCHLD   18) SIGCONT    19) SIGSTOP     20) SIGTSTP  
21) SIGTTIN    22) SIGTTOU   23) SIGURG     24) SIGXCPU     25) SIGXFSZ  
26) SIGVTALRM  27) SIGPROF   28) SIGWINCH   29) SIGIO       30) SIGPWR  
31) SIGSYS     34) SIGRTMIN  35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3  
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8  
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13  
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12  
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7  
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2  
63) SIGRTMAX-1 64) SIGRTMAX  
ezzouhri.A@janati_idrissi.M:~$
```

(b) Fermer le processus 'essai' en utilisant Le signal TERM (SIGTERM). (c) forcer la fermeture d'un processus, on peut utiliser le signal "KILL" (SIGKILL).

```
tp_signal.c x  
#include <stdio.h>  
#include <signal.h>  
#include <stdlib.h>  
#include <string.h>  
  
void main(int argc, char *argv[]){  
    int nbs;  
    nbs=atoi(argv[1]);  
    if(signal(SIGINT, SIG_IGN) == SIG_ERR)  
{  
        perror("SIGINT");  
        exit(1);  
    }  
  
    if(signal(SIGTERM, SIG_IGN) == SIG_ERR){  
        perror("SIGTERM");  
        exit(1);  
    }  
    sleep(nbs);  
    exit(1);  
}
```

```

janati@ubuntu: ~
ezzouhri.A@janati_idrissi.M:~$ gcc tp_signal.c -o ver
ezzouhri.A@janati_idrissi.M:~$ ./ver 30 &
[1] 13162
ezzouhri.A@janati_idrissi.M:~$ ps
  PID TTY          TIME CMD
 2696 pts/1        00:00:05 bash
13162 pts/1        00:00:00 ver
13163 pts/1        00:00:00 ps
ezzouhri.A@janati_idrissi.M:~$ kill -TERM 13162
ezzouhri.A@janati_idrissi.M:~$ ps
  PID TTY          TIME CMD
 2696 pts/1        00:00:05 bash
13166 pts/1        00:00:00 ps
[1]+  Exit 1                  ./ver 30
ezzouhri.A@janati_idrissi.M:~$

```

(d) Quelles sont les commandes équivalentes à la commande `kill -9 1414` ?

Les commandes sont : `kill -SIGKILL 1414`

### Exercice2 :

(a) Ecrire une fonction nommée 'fonc\_react' qui réagit à un signal qui est passé du programme principal 'main' dans le seul paramètre sig de 'fonc\_react' . Cette fonction est appelée par le programme principal, lorsqu'un signal arrive (Ctrl+c). Le main tourne dans une boucle infinie et, imprime un message une fois par seconde. Le rôle de la fonction 'fonc\_react', est d'imprimer un message (de réussite du captage du signal) et initialise par défaut le signal en question en composant Ctrl+c.

```

tp5_exercice2.c x
#include<signal.h>
#include<stdio.h>
#include<unistd.h>

void fonc_react(int sig){
printf("moi fonc_react ! , j'ai rattrapé un signal %d \n",sig);

(void)signal(SIGINT,SIG_DFL);
}

int main(){
(void)signal(SIGINT,func_react);
while(1){
printf("HELLO WORLD !! \n");
sleep(1);
}}

```

```

janati@ubuntu: ~
ezzouhri.A@janati_idrissi.M:~$ gedit tp5_exercice2.c
ezzouhri.A@janati_idrissi.M:~$ gcc tp5_exercice2.c -o ver
ezzouhri.A@janati_idrissi.M:~$ ./ver
HELLO WORLD !!
HELLO WORLD !!
HELLO WORLD !!
HELLO WORLD !!
^Cmoi fonc_react ! , j'ai rattrapé un signal 2
HELLO WORLD !!
HELLO WORLD !!
HELLO WORLD !!
^C
ezzouhri.A@janati_idrissi.M:~$

```

(b) Que remarquez-vous ?

On remarque que ctrl+c ne fait pas son travail dès la première fois, c'est la deuxième qui interrompt le programme.

### Exercice3 :

(a) Ecrire un programme C nommé 'sigfunc' dont son exécution est infinie avec des arrêts périodiques d'une seconde. Si on veut arrêter l'exécution de ce programme par Ctrl + C, comment peut-on gérer cette action en utilisant la fonction signal ? (b) affichez un message d'erreur au niveau du 'main' si le signal n'est pas capté. la fonction signal utilise comme 2ème paramètre 'sig\_handler' qui correspond au nom d'une fonction définie dans ledit programme. (c) Utilisez cette fonction pour afficher sur la sortie standard si le signal a été reçu ou pas.

```
tp5_exercice3.c x
#include<signal.h>
#include<stdio.h>
#include<unistd.h>

void sig_handler(int signo){
if(signo == SIGINT)
printf("SIGINT reçu .\n");
}

int main(void){
if(signal(SIGINT,sig_handler) == SIG_ERR)
printf("je ne peut pas capter SIGINT \n");
while(1)
sleep(1);
return 0;
}
```

```
janati@ubuntu: ~
ezzouhri.A@janati_idrissi.M:~$ gedit tp5_exercice3.c
ezzouhri.A@janati_idrissi.M:~$ gcc tp5_exercice3.c -o ver
ezzouhri.A@janati_idrissi.M:~$ ./ver
^CSIGINT reçu .
^CSIGINT reçu .
^CSIGINT reçu .
^CSIGINT reçu .
^CSIGINT reçu .
^CSIGINT reçu .
^CSIGINT reçu .
^Z
[25]+ Stopped ./ver
ezzouhri.A@janati_idrissi.M:~$
```

### Exercice4 :

Apportez les modifications nécessaires au programme 'sigfunc' afin de pouvoir utiliser des signaux définis (signal USR1 par exemple) par l'utilisateur. (b)Apportez les modifications nécessaires à la fonction 'sig\_handler' afin de pouvoir vérifier la ou la non réception des signaux KILL et STOP.

```
tp5_exercice4.c x
#include<signal.h>
#include<stdio.h>
#include<unistd.h>

void sig_handler(int signo){
    if(signo == SIGUSR1)
        printf("SIGUSR1 reçu.\n");
    else if(signo == SIGKILL)
        printf("SIGKILL reçu.\n");
    else if(signo == SIGSTOP)
        printf("SIGSTOP reçu.\n");
}

int main(){
    if (signal(SIGUSR1,sig_handler) == SIG_ERR)
        printf("\nje ne peux pas intercepter SIGUSR1\n");
    if(signal(SIGKILL,sig_handler) == SIG_ERR)
        printf("\nje ne peux pas intercepter SIGKILL\n");
    if(signal(SIGSTOP,sig_handler) == SIG_ERR)
        printf("\nje ne peux pas intercepter SIGSTOP\n");
    while(1)
        sleep(1);
    return 0;
}
```

```
janati@ubuntu: ~
ezzouhri.A@janati_idrissi.M:~$ gedit tp5_exercice4.c
ezzouhri.A@janati_idrissi.M:~$ gcc tp5_exercice4.c -o ver
ezzouhri.A@janati_idrissi.M:~$ ./ver&
[26] 8985
ezzouhri.A@janati_idrissi.M:~$
je ne peux pas intercepter SIGKILL

je ne peux pas intercepter SIGSTOP
kill -SIGUSR1 8985
ezzouhri.A@janati_idrissi.M:~$ SIGUSR1 reçu.
```

(c) que remarquez-vous ?

On constate que les deux signaux kill et stop ne sont pas capter , au contraire ,le signal SIGUSR1 est reçu.