



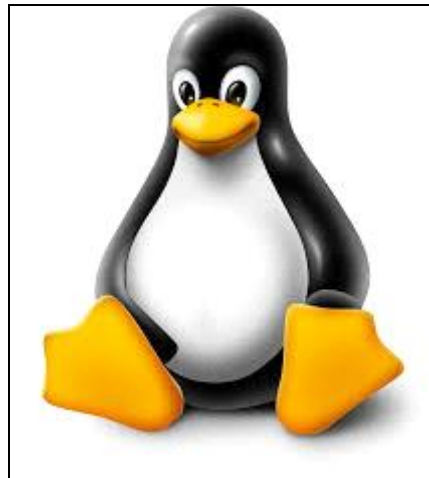
جامعة سيدي محمد بن عبد الله  
كلية العلوم ظهر المهرارز

جامعة سيدي محمد بن عبد الله  
كلية العلوم ظهر المهرارز

Université Sidi Mohamed Ben Abdellah  
Faculté des Sciences Dhar Mahraz



Compte Rendu De Travaux Pratique  
System d'Exploitation  
Travaille Pratique 5 : Les Signaux



SMI – S4

Réaliser par :

Prénom :	CNE :	Nom :
Achraf	1513755449	Tazi
Oussama	1311778906	Nadrani

## Exercice1 :

*(a)Affichez le nom et les numéros des signaux.(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).(d) Quelles sont les commandes équivalentes à la commande kill -9 1414 ?*

❖ (a) :

```
nadrani@nadrani-pc:~$ 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
```

(b) : D'abord lancer un processus (car notre système linux ne contient pas du processus qui s'appelle `essai`, donc on doit utiliser Firefox)

Ensuite on doit afficher le numéro PID du processus par la commande `Pidof` ( le processus utilisé ).

Puis on utilise l'un des commandes suivantes pour fermer ce processus

**Fermeture normal : en utilisant la commande `kill -SIGTERM`**

❖ (c) : Forcer la fermeture :en utilisant la commande **`kill -SIGKILL`**

❖ (d) : **`kill -s SIGKILL pid_processus`**  
**`kill -s kill pid_processus`**

## **Exercice2 :**

- (a) *Ecrire une fonction nommée 'func\_react' qui réagit à un signal qui est passé du programme principal 'main' dans le seul paramètre sig de 'func\_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 'func\_react', est d'imprimer un message (de réussite du captage du signal) et initialise par défaut le signal en question en composant Ctrl+c. (b) Que remarquez-vous ?*

❖ (a) :code->

```
#include<stdio.h>
#include<stdlib.h>
#include<signal.h>
#include<string.h>
void func_react(int sig)
{
    printf("\nle signal %d a etait detecter\n",sig);
    (void)signal(SIGINT,SIG_DFL);
}

void main(int argc, char *argv[]){
    (void)signal(SIGINT,func_react);
    while(1){
        printf("\nTAZI NADRANI\n");
        sleep(1);
    }
}
```

## Résultat->

```
nadrani@nadrani-pc:~/sys_tp/TpSys5$ gcc ex1.c -o TP5_1
nadrani@nadrani-pc:~/sys_tp/TpSys5$ ./TP5_1

TAZI NADRANI

TAZI NADRANI

TAZI NADRANI
^C
le signal 2 a etait detecter

TAZI NADRANI

TAZI NADRANI
^C
nadrani@nadrani-pc:~/sys_tp/TpSys5$
```

**Remarque->** Le programme n'est arrêté qu'après un 2ème Control-c.

## 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.*

❖ code->



## Exercice4 :

- (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.*

```
#include<stdio.h>
#include<unistd.h>
#include<signal.h>

void sig_handler(int sig){
    if(sig == SIGUSR1){
        printf("signal reçu \n");
    }
    if(sig == SIGKILL){
        printf("signal reçu\n");
    }
    if(sig == SIGSTOP){
        printf("signal reçu \n");
    }
}

int main(int argc, char *argv[]){
    if(signal(SIGUSR1, sig_handler)==SIG_ERR)
        printf("erreur SIGUSR1");
    if(signal(SIGKILL, sig_handler)==SIG_ERR)
        printf("erreur SIGKILL");
    if(signal(SIGSTOP, sig_handler)==SIG_ERR)
        printf("erreur SIGSTOP");
    while(1)
        sleep(1);
    return 0;
}
```

(b)  erreur SIGKILL  
erreur SIGSTOP

(c) : *Donc la sortie ci-dessus indique clairement que, dès que l'appel système «signal» essaie d'utiliser les signaux KILL et STOP par sig\_handler, la fonction signal échoue indiquant que ces deux signaux ne peuvent pas être pris.*