

TD Programmation Système

Série 2 Signaux : Correction

Exercice 1

- On passe en paramètre le nombre de seconde où le signal SIG_IGN sera ignoré
- On ignore le signal SIG_IGN
- On retourne au comportement par défaut

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

int main(int argc, char *argv[]){

    int NB;
    NB=atoi(argv[1]);
    /*on ignore le signal SIGINT(ctr^c)
    signal(SIGINT, SIG_IGN);
    sleep(NB);

    /* On remet le comportement par Défaut de SIGINT */
    signal(SIGINT, SIG_DFL);
    return 0 ;
}
```

Exercice 2

On vous demande d'écrire deux programmes

- "Boucle.c" qui redéfinit le signal passé en paramètre et qui boucle après avoir affiché son pid
- "Envoi.c" qui récupère un pid et un signal passés en paramètre et qui envoie le signal au processus identifié par pid

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

void afficher (int sig){
    printf("numero du signal : %d\n", sig);
}

int main(int argc, char *argv[]){

    int sig, pid;
    sig=atoi(argv[1]);

    signal(sig, afficher);
    printf("pid=%d\n", getpid());
    while(1);
    return 0 ;
}
```

Boucle.c

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

int main(int argc, char *argv[]){
    int sig, pid;

    sig=atoi(argv[1]);
    pid=atoi(argv[2]);
    kill(pid, sig);

    return 0 ;
}
```

Envoi.c

Exercice 3: Synchronisation Père/Fils

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

void Handler(int sig)
{    } /* handler sans comportement */

int main(){
int pid,n=0;

signal(10,Handler);/* Redéfinition du Signal SIGUSR1*/
pid=fork();

if(pid==-1){
    perror("fork");
    exit(0);
}

if (pid==0){
    while(n<100){
        printf("fils:");
        do { n=n+2;
            printf(" %d",n);

            }while (n%5!=0);
        printf("\n");
        kill(getppid(),10);
        pause();
    }
}
else{
    while(n<99){
        if (n==0) pause();
        printf("pere:");
        do { n=n+3;
            printf(" %d",n);

            }while (n%5!=0);
        printf("\n");
        kill(pid,10);
        pause();
    }
}
return 0 ;
}
```