TP 2
Les signaux

Gestion des processus
Semestre 3

1 Exercice 1

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

typedef void ( *PtrFct) (int);

void traiterSIGUSR1(int sig);
int main (int argc, char** argv) {
  PtrFct retFct ;
  retFct=signal( SIGUSR1, traiterSIGUSR1 );

  if (retFct == SIG_ERR) {
    perror("echec signal");
    exit(1);
  }

  while (42) {
    sleep(5);
  }
}
void traiterSIGUSR1(int sig) {
  switch ( sig ) {
    case SIGUSR1 : printf("PID= %d\n", getpid() );
      printf("n° du signal recu = %d\n", sig );
      break;
    default :    printf("\n Erreur système !!!!\n");
  }
  exit(2);
}
```

Listing 1 – Exercice 1 – signal1.c

2 Exercice 2

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

int main(int nbParam, char** tabParam) {
  int err;
  pid_t pid;

  switch (nbParam) {
     case 2 :
      sscanf(tabParam[1], "%d", &pid);
      printf("envoi du signal SIGUSR1 au processus= %d\n", pid);
      err=kill( pid, SIGUSR1 );

      if (err == -1) {
        perror("echec kill");
        exit(1);
      }
    break;
    default :
      printf("***nombre de parametres incorrect!!!\n");
      exit(1);
   }
}
```

Listing 2 – Exercice 2 – signal2.c

3 Exercice 3

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

#define TIMEOUT1 1
typedef void(* PtrFct)(int);

void timeout(int sig);
jmp_buf ptRep; /* doit etre declare en variable globale */

int main (int argc, char** argv) {
  PtrFct retFct;
  int ret;
  int nbEssai=0;
  unsigned duree=10; /* delai d'attente 10 sec */
  char mess[257];

  retFct = signal(SIGALRM, timeout);
  if (retFct == SIG_ERR) {
    perror("echec signal");
```

```
24      exit(1);
25    }
26    ret = setjmp(ptRep);
27    if (ret == TIMEOUT1 ) {
28       if (nbEssai > 2 ) {
29         printf("echec saisie!!!\n");
30         exit(2);
31       }
32     }
33    nbEssai++;
34    printf("entrer votre message:\n");
35    alarm(duree);
36    fgets(mess,256,stdin);
37    alarm(0);
38    printf("saisie OK !!!\n");
39  }
40  void timeout(int sig) {
41    switch (sig) {
42      case SIGALRM : printf("n° du signal recu = %d\n", sig );
43        sigrelse(SIGALRM);        /* demasquer le signal SIGALRM */
44        longjmp(ptRep,TIMEOUT1);
45      default :
46        printf("\n Erreur système !!!!\n");
47        exit(3);
48    }
49  }
```

Listing 3 – Exercice 3 – signal3.c