TP 1
Gestion des processus

Antoine de Roquemaurel et Mathieu Soum
Semestre 3

1 Exercice 1

```
drn0073a@r-info-jade-l03:~$
Appuyez sur ENTRÉE ou tapez une commande pour continuer

mardi 20 septembre 2011, 10:06:52 (UTC+0200)
Le login est satenske
L'IUD est 1000
-----------------------------------------


Processus fils 1. son PID est 2900
Processus fils 2. son PID est 2901
fin anormale fils PID=2901, numéro du signal=11
fin normale du PID fils=2900, statut=3
```

Listing 1 – Trace du programme proc1: **proc1.trace**

```
/*************************************************
*  ASR => Gpr3                                   *
*************************************************
*  Login : drn0073a                              *
*                                                *
*************************************************
*  Groupe: B                                     *
*                                                *
*************************************************
*  Nom-prénom : de ROQUEMAUREL Antoine           *
*  Nom-prénom : SOUM Mathieu                     *
*
*************************************************
*  TP n 1                                       *
*************************************************
*  Nom du fichier : proc1.c                      *
*************************************************/

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

#define OK 0
#define PAS_OK -1

```

```c
void traitement_fils1(void);
void traitement_fils2(void);

int main(int argc, char *argv[]) {
  pid_t idFils1, idFils2, pidPere, idProcRetour;
  int rapport, numSig, statut;

  pidPere = getpid();

  system("date");
  system("echo Le login est $USER\n");
  printf("L'IUD est %d\n", getuid());
  printf("-----------------------------------------\n\n\n");

  idFils1 = fork(); // On créer le premier fils
  if (pidPere == getpid())  // Si c'est le père, on créé un deuxième fils!
    idFils2 = fork();

  switch ( idFils1 ) {
    case PAS_OK :
      perror("Erreur de lancement du processus fils1");
      exit(1);
    case OK :
      traitement_fils1();
      break;
  }
  switch ( idFils2 ) {
    case PAS_OK:
      perror("Erreur de lancement du processus fils2");
      exit(1);
    case OK :
      traitement_fils2();
      break;
  }

  idProcRetour = wait(&rapport);
  // tant que le processus n'est pas arrêté
  while (idProcRetour != -1) {
    numSig = rapport & 0x7F;

    switch (numSig) {
      case 0 :
        statut = (rapport >> 8) & 0xFF;
        printf("fin normale du PID fils=%d, statut=%d\n", idProcRetour,
            statut);
        break;
      default:
        printf("fin anormale fils PID=%d, numéro du signal=%d\n",
            idProcRetour, numSig);
        break;
    }
    idProcRetour = wait(&rapport);
  }

  return 0;
}

// processus fils 1
```

```
83 void traitement_fils1(void) {
84   printf("Processus fils 1. son PID est %d\n", getpid());
85   exit(3);
86 }
87
88 // processus fils 2
89 void traitement_fils2(void) {
90   int* ptr = NULL;
91   printf("Processus fils 2. son PID est %d\n", getpid());
92
93   *ptr = 42;
94   exit(OK);
95 }
```

Listing 2 – Code source de **proc1.c**

2 Exercice 2

```
1  drn0073a@r-info-jade-l03:~$
2  Appuyez sur ENTRÉE ou tapez une commande pour continuer
3  mardi 20 septembre 2011, 10:17:44 (UTC+0200)
4  Le login est satenske
5  L'IUD est 1000
6  -----------------------------------------
7
8
9  Ici le processus fils 2 de PID 3458 !
10 Ici le processus fils 1 de PID 3457 !
11 La chaine passée en paramètre est : Cacamou
12 Le paramètre en décimal est 42(10) et en hexadécimal 2a(16) !
13 fin normale du PID fils=3457, statut=0
14 fin normale du PID fils=3458, statut=0
```

Listing 3 – Trace du programme proc2: **proc2.trace**

```
1  /*************************************************
2    *  ASR => Gpr3                                  *
3    *************************************************
4    *  Login : drn0073a                             *
5    *                                               *
6    *************************************************
7    *  Groupe: B                                    *
8    *                                               *
9    *************************************************
10   *  Nom-prénom : de ROQUEMAUREL Antoine     *
11   *  Nom-prénom : SOUM Mathieu                    *
12   *                                               *
13   *                                               *
14   *************************************************
15   *  TP n: 1                                     *
16   *************************************************
17   *  Nom du fichier : proc2.c                     *
18   *************************************************/
19
20 #include <stdlib.h>
21 #include <stdio.h>
22 #include <sys/types.h>
```

```c
#include <unistd.h>

#define OK      0
#define PAS_OK -1

int main(int argc, char *argv[]) {
  pid_t idFils1, idFils2, pidPere, idProcRetour;
  int rapport, numSig, statut, err1, err2;
  char* argP2fils2[] = { "main", "42", NULL };

  pidPere = getpid();

  system("date");
  system("echo Le login est $USER et l UID $UID\n");
  system("echo ----------------------------------------\n");
  printf("\n\n");

  idFils1 = fork(); // On créer le premier fils
  if (pidPere == getpid()) { // Si c'est le père, on créé un deuxième fils!
    idFils2 = fork();
  }

  switch ( idFils1 ) {
    case PAS_OK :
      perror("Erreur de lancement du processus fils1");
      exit(1);
      break;
    case OK :
      err1 = execl("p2fils1", "main", "une super chaine de caracteres", NULL
          );
      if (err1 == -1) {
        perror("Erreur au niveau du execl");exit(2);
      }
      break;
  }
  switch ( idFils2 ) {
    case PAS_OK:
      perror("Erreur de lancement du processus fils2");
      exit(1);
      break;
    case OK :
      err2 = execvp("p2fils2", argP2fils2);
      if (err2 == -1) {
        perror("Erreur au niveau du execvp");exit(2);
      }
      break;
  }

  idProcRetour = wait(&rapport);
  // tant que le processus n'est pas arrêté
  while (idProcRetour != -1) {
    numSig = rapport & 0x7F;

    switch (numSig) {
      case 0 :
        statut = (rapport >> 8) & 0xFF;
        printf("fin normale du PID fils=%d, statut=%d\n", idProcRetour,
            statut);
```

```
79            break;
80          default:
81            printf("fin anormale fils PID=%d, numéro du signal=%d\n",
                 idProcRetour, numSig);
82            break;
83        }
84        idProcRetour = wait(&rapport);
85      }
86
87      return 0;
88    }
```

Listing 4 – Code source de **proc2.c**

```
1   /*************************************************
2   *  ASR => Gpr3                                   *
3   *************************************************
4   *  Login : drn0073a                              *
5   *                                                *
6   *************************************************
7   *  Groupe: B                                     *
8   *                                                *
9   *************************************************
10  *  Nom-prénom : de ROQUEMAUREL Antoine           *
11  *  Nom-prénom : SOUM Mathieu                     *
12  *                                                *
13  *                                                *
14  *************************************************
15  *  TP n: 1                                      *
16  *************************************************
17  *  Nom du fichier : p2fils1.c                      *
18  *************************************************/
19
20  #include <stdlib.h>
21  #include <stdio.h>
22  #include <sys/types.h>
23  #include <unistd.h>
24
25  int main(int argc, char** argv) {
26    printf("Ici le processus fils 1 de PID %d !\n", getpid());
27    printf("La chaine passée en paramètre est : %s\n", argv[1]);
28    exit(0);
29
30    return 0;
31  }
```

Listing 5 – Code source de **p2fils1.c**

```c
/*********************************************
*  ASR => Gpr3                                *
*********************************************
*  Login : drn0073a                           *
*                                             *
*********************************************
*  Groupe: B                                  *
*                                             *
*********************************************
*  Nom-prénom : de ROQUEMAUREL Antoine        *
*  Nom-prénom : SOUM Mathieu                  *
*                                             *
*                                             *
*********************************************
*  TP n: 1                                   *
*********************************************
*  Nom du fichier : p2fils2.c                 *
*********************************************/

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

int main(int argc, char** argv) {
  printf("Ici le processus fils 2 de PID %d !\n", getpid());
  printf("Le paramètre en décimal est %d(10) et en hexadécimal %x(16) !\n",
      atoi(argv[1]), atoi(argv[1]));
  exit(0);

  return 0;
}
```

Listing 6 – Code source de **p2fils1.c**