

### 3 Synthèse : processus communiquant par signaux et tubes

On considère le programme suivant :

```

1 #include <stdio.h>
2 #include <unistd.h>
3 #include <sys/types.h>
4 #include <signal.h>
5 #include <stdlib.h>
6
7 #define NTESTS 5
8 static int PairImpair = -1;
9 static int Iboucle = 0;
10 static int PrintM1 = 1;
11 static int p[2];
12
13 void H1 (int sig) {
14     if (PrintM1==1) {
15         /* PrintM1 indique simplement s'il s'agit de la première activation de H1(_) */
16         printf ("Message_1, X1=%d, a_recu_Y1=%d\n", (int) getpid(), sig);
17         PrintM1 = 0;
18     }
19     else {
20         if (sig==SIGQUIT) {
21             printf ("Message_2, X2=%d, a_recu_Y2=%d\n", (int) getpid(), sig);
22             exit(3);
23         }
24     }
25     /* Iboucle<0 Iboucle=Iboucle; /* Iboucle peut devenir < 0, en cas de dépassement */
26     /* PairImpair=1 si Iboucle est impair et =0 sinon */
27     PairImpair=Iboucle%2;
28     printf ("Iboucle=%d, PairImpair=%d\n", Iboucle, PairImpair);
29     write(p[1], &PairImpair, sizeof(int));
30     Iboucle=0;
31 }
32
33 void H2 (int sig) {
34     printf ("Message_3, X3=%d, a_recu_Y3=%d\n", (int) getpid(), sig);
35 }
36
37 int main (int argc, char *argv[]) {
38     int pid1, pid2, pid3, ret, i, n;
39     int q[2];
40     float nbtotal, nbpair;
41     float bilan, nbpairmoyen;
42
43     printf ("Message_4, X4=%d, a_recu_Y4=%d\n", (int) getpid(), getppid());
44
45     pipe(p);
46     pipe(q);
47     signal(SIGQUIT, H1);
48     signal(SIGUSR1, H1);
49     signal(SIGUSR2, H2);
50     pid1=fork();
51     if (pid1 != 0) {
52         printf (".....pid1=%d\n", (int) pid1);
53         pid2=fork();
54         if (pid2 != 0) {
55             close(p[1]);
56             close(q[1]);
57             close(p[0]);
58
59             printf (".....pid2=%d\n", (int) pid2);
60             kill(pid1, SIGUSR1);
61             sleep(1);
62             kill(pid1, SIGUSR1);
63
64             kill(pid2, SIGUSR2);
65             ret=wait(&n);
66             if (WIFSIGNALED(n))
67                 printf ("Message_5, X5=%d, a_recu_Y5=%d\n", ret, WTERMSIG(n));
68             else
69                 printf ("Message_6, X6=%d, a_recu_Y6=%d\n", ret, WEXITSTATUS(n));
70             kill(pid1, SIGINT);
71             ret=wait(&n);
72             if (WIFSIGNALED(n))
73                 printf ("Message_7, X7=%d, a_recu_Y7=%d\n", ret, WTERMSIG(n));
74             else
75                 printf ("Message_8, X8=%d, a_recu_Y8=%d\n", ret, WEXITSTATUS(n));
76             bilan=999;
77             ret=read(q[0], &bilan, sizeof(float));
78             if (ret>0) {bilan = bilan * 2;}
79             printf (" _bilan=%f\n", bilan);
80         }
81     }
82     else { /* pid2 == 0 */
83         close(p[1]);
84         close(q[1]);
85         printf ("Message_9, X9=%d, a_recu_Y9=%d\n", (int) getpid(), getppid());
86         pause();
87         execp("ps", "ps", NULL);
88         ret=wait(&n);
89         /* Ligne 87*/
90     }
91     else { /* pid1 == 0 */
92         pid3=fork();
93         if (pid3 != 0) {
94             close(q[1]);
95             printf (".....pid3=%d\n", (int) pid3);
96             printf ("Message_10, X10=%d, a_recu_Y10=%d\n", (int) getpid(), getppid());
97             for (;
98                 Iboucle ++;
99                 /* Ligne 99 */
100             )
101                 else /* pid3 == 0 */
102                 {
103                     close(p[1]);
104                     close(q[0]);
105                     nbpair = 0; nbtotal=0; nbpairmoyen = -1; nbtotal=0;
106                     while ( read (p[0], &i, sizeof(int)) > 0 ) /* Ligne 106*/
107                     {
108                         nbtotal = nbtotal + 1;
109                         if (i==0) nbpair = nbpair+1;
110                     }
111                     if (nbtotal !=0) nbpairmoyen = nbpair / nbtotal;
112                     printf ("nbpair=%f\n", nbpair);
113                     printf ("nbpairmoyen=%f\n", nbpairmoyen);
114                     write (q[1], &nbpairmoyen, sizeof(float));
115                 }
116             }
117             return 0;
118         }
119     }

```

## Questions

Une exécution du code exam avec `NBTESTS=5` donne la sortie suivante (notez que certaines valeurs ont volontairement été remplacées par des `????`)

```
Message 4, X4=???? de pere Y4=????
..... pid1 = 2504
..... pid2 = 2505
..... pid3 = 2506
..... pid4 = 2507
Message 10, X10=???? de pere Y10=????
Message 9, X9=???? de pere Y9=????
Message 1, X1=???? a recu Y1=10
Iboucle=241534614 PairImpai=0
Iboucle=242681100 PairImpai=0
Iboucle=485825588 PairImpai=0
Iboucle=2428202438 PairImpai=0
Message 3, X3=???? a recu Y3=????
Iboucle=485866617 PairImpai=1
PID TTY TIME CMD
2130 pts/0 00:00:00 bash
2503 pts/0 00:00:00 exam
2504 pts/0 00:00:04 exam
2505 pts/0 00:00:00 ps
2506 pts/0 00:00:00 exam
Message 6, X6=???? et Y6=????
nbpairs=4,00000
nbpairsym=0,800000
Message 7, X7=???? et Y7=????
bilan=1,60000
```

1. Décrire l'architecture de communication entre processus et indiquer/représenter les descripteurs ouverts par processus.
2. Quels sont les descripteurs connectés au processus qui effectue la boucle infinie (Ligne 96). Comment ce processus sort-il de cette boucle?
3. Compléter et expliquer l'affichage des valeurs `Xi` et `Yi` pour  $i=1,10$  Expliquer en particulier pourquoi les messages 2, 5 et 8 ne sont pas affichés.
  - (a) on trouve 3 lignes dont le motif est  
..... pts/O ..... exam ?
  - (b) Quel que soit l'ordre de séquençement des processus par le système trouvera-t-on toujours exactement 3 lignes avec  
..... pts/O ..... exam ?
5. Expliquer comment le processus qui effectue le `while` en Ligne 106 sort de cette boucle.
6. Si on supprime le `close(p1)` en ligne 82, que se passe-t-il?
7. Si on supprime le `close(q1)` en ligne 93, que se passe-t-il?
8. Que se passe-t-il si on supprime le `close(p1)` en Ligne 55?
9. Que se passe-t-il si on déplace `close(q1)` en Ligne 56 après le `wait` en Ligne 65?
10. Si on supprime l'instruction `kill(pid2, SIGUSR2)` en Ligne 64, que se passe-t-il?
11. Dans quel cas atteint-on la ligne 87?
12. Si on ajoute l'instruction `signal(SIGUSR1, SIG_IGN)` avant la boucle infinie ligne 96 que se passe-t-il? Indiquer la valeur de bilan.
13. Que se passe-t-il si on supprime la ligne 70 (`kill(pid1, SIG_INT)`) ?
14. Que se passe-t-il si on remplace `kill(pid1, SIGINT)` en ligne 70 par `kill(pid1, SIGQUIT)`? Indiquer notamment l'effet sur l'affichage du message 7.
15. Pour de grandes valeurs de `NBTESTS`, vers quelle valeur doit tendre la variable bilan?