TP 01 – Parallélisme et synchronisation

Exercice 1. *`fork` et `wait`*

Qu'affichent à l'écran les programmes suivants (donner une possibilité) ?

1. ```
   #include <stdio.h>
   #include <stdlib.h>
   #include <string.h>
   int main(void){
       fork();
       fork();
       fork();
       printf("hi");
       return 0;
   }
   ```

2. ```
   #include <stdio.h>
   #include <sys/types.h>
   #include <unistd.h>
   int main(void){
     int i=0;
     for(i=0;i<4;i++){
       fork( );
     }
     printf("ho");
     return 0;
   }
   ```

3. ```
   #include <stdio.h>
   #include <sys/types.h>
   #include <unistd.h>
   int main(void){
     fork( );        printf("hee");
     fork( );        printf("ha");
     fork( );        printf("ho");
     return 0;
   }
   ```

Exercice 2. *`fork` bomb*

✎ Ecrire une *fork bomb* ! C'est attaque de type *denial of service*.
Si vous êtes courageux, lancez-la ! !
   ```

Exercice 3.

Implémenter les généalogies suivantes, chacun doit écrire à l'écran le message :

`Je m'appelle <mon_nom> de numéro <getpid()> et de parent <getppid()>.`

Puis attendre une seconde (fonction `sleep` de la bibliothèque `unistd.h`).

Chacun attend la mort de ses enfants puis exécute l'instruction `exit(0)`.

A chaque mort d'un de ses enfants, le parent pleure et affiche :

`Je suis <mon_nom> et je pleure car <pid_du_mort> est mort et je l'aimais bien.`

1. Alice a deux enfants, Bob et Bob. Bob et Bob sont identiques, ils ont chacun deux enfants prénommés Charly et David. Les deux Charly et les deux David n'ont pas d'enfant.
2. Emilie a un enfant, François, qui a un enfant, George, qui a trois enfants, Henri, Isabelle et Jérôme. Henri a un fils Kader, Isabelle a une fille Louise, et Jérôme a une fille Manon qui a une fille Nora. Kader, Louise et Nora n'ont pas d'enfant.

Exercice 4. *armée de zombis*

✎ Ecrire un programme créant une armée de $10\,000$ zombis !
Les compter avec une commande Bash combinant `ps` et `wc`.

Exercice 5. *signaux : drame familial*

✎ En utilisant les fonctions `fork` et `kill`, écrire le programme C correspondant à l'histoire : un père a deux fils qui entrent dans des boucles infinies, il tue ses deux fils et se suicide.

Exercice 6. *signaux : handler*

✎ Ecrire un handler de signaux pour qu'un processus puisse recevoir des signaux `SIGINT` et écrire `ok` à chaque réception. Tester.

Exercice 7. *tube de communication*

1. Utiliser un tube de façon à ce qu'un processus père puisse envoyer des chiffres à son fils, et ce dernier les additionne et écrit `multiple de 3` à chaque fois que la somme est un multiple de 3. Les chiffres envoyés par le père seront lus sur son entrée standard (le clavier), qui sera recopiée dans le tube en combinant une redirection de sa sortie standard vers le tube et un recouvrement par la commande bash `cat`.
2. Utiliser trois processus P_0, P_1 et P_2 et trois tubes de façon à ce que P_i communique avec $P_{i+1 \mod 2}$, et chaque processus affiche ce qui lui arrive dans le tube où il lit et recopie ce qu'il a lut dans le tube où il écrit. Le processus P_0 doit initialement écrire bla dans le tube.