30/01/2018
Reseau

① TD

Séance 2 - TD - Réalisation des processus - Signaux.

Exercice 1 -

On redéfinit le handler du signal SIGINT, puis on attend à l'infini un control C. (n'importe quel signal)

Exercice 2 -

Le même que le pid du processus père.

Exercice 3 -

On kill les processus du même groupe que pid, si pid est le père, sinon on attend.

Exercice 4 -

On kill le processus désigné par pid si pid est le père, sinon on attend.

Exercice 5 -

```
void handler (int SIG) { }

int attend (int t){
    int a;
    signal (SIGINT , handler);
    printf ("%i est le temps passé à attendre ", sleep(t));
    exit (0);
}
```

Exercice 6 -

```
void handler (int sig) {
   static int counter = 0;
   printf ("bip");
   alarm(1);
   if (c == 6) {
       printf ("bye");
       exit (0);
   }
   c++;
}
```

```
int main (void) {
    signal (SIGALRM, handler);
    while (1) {
        alarm (1);
    }
return 0;
}
```

Exercice 7 :

Il redefinit le handler du signal SIGUSR2, puis fork. Le fils envoie 5 signaux SIGUSR2 au handler et print 5 fois le message.

La valeur peut être entre 2 et 5 en fonction de la perte de signal potentiel.

Exercice 8 :

c dur

06/02/2018
SYSTEME

(1) (TD) TD n°3 : Fichier, flots d'entrées - sorties et tubes

1. Redirection des entrées - sorties

Exercice 1- > redirige en écrasant
>> redirige en concatenant à la fin
ls -l > toto ll dans toto

Exercice 2- met l'entrée standad dans toto

Exercice 3- ls -l >> toto concatene à la fin.

Exercice 4- ça tri le contenu du fichier fournit et l'affiche à l'écran.

Exercice 5- c'est pareil, ça crée toto.

Exercice 6- less sur les fichiers affiché par ls -l. (les lignes)

Exercice 7- find . -type f -print | wc -l - on cherche les fichiers du ./ et on compte le nombre de ligne.

2 - Descripteur de fichiers

Exercice 8-

```
int main (int argc, char * []){
    int retour
    char

    retour = open (argv[1], O_RDONLY);
```

Exercice 9 -

fd2 = un int correspondant au fichier ouvert (numero de descripteur)
3

Exercice 10 -

fd2 = pareil mais 4

Exercice 11 -

c = l

Exercice 12 -

c = a

Exercice 13 -

c = a

4 - Manipulation de tubes -

4.1 - Ça a l'air bien.

4.2

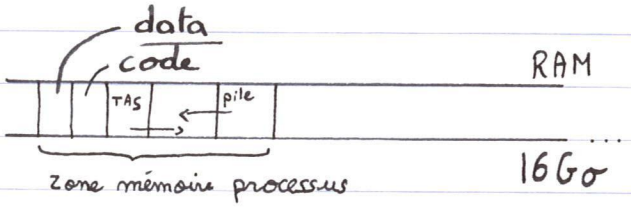
23/01/2018

SYSTEME

① CM

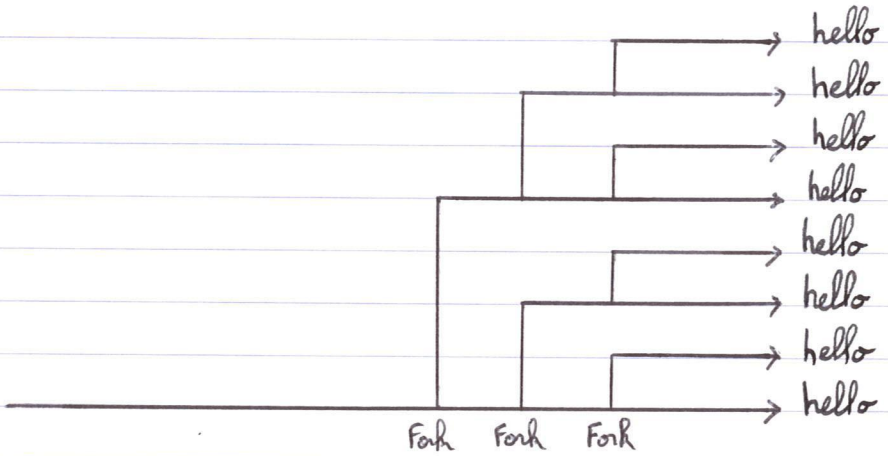
varia.marangozova@imag.fr

1 - Création de processus



Exercice 1-
En fonction de la valeur de retour de Fork();, on aura soit child : x = 2 ou parent: x = 0

Exercice 2-



-> 8 hellos.

Exercice 3- 1er : 4
2eme : 8

Exercice 4- fait en TP.

```
int numpro (int n){
  int i
  for (i=0; i<n; i++){
    if(!fork()){
      exit(0);
    }
  }
}
```

2 - Relations entre processus pere et fils.

Exercice 5 -

premier for : exit (pid)
premier while : wait pid (pid ...)

Exercice 6 -

6435 lignes, leurs ordres d'apparition peuvent changer, rien n'est executés en même temps et le pere n'est pas forement fini avant (executé)
-> le programme ne finira pas. il va wait à l'infini.

Exercice 7 -

il devrait lister le contenu du /.