

# Programmation Système - TD 2

23/10/20

## 1 Exercice 1 : fork()

Programmer l'exemple du cours *fork.c* (diapositive 6).

Modifier le programme pour passer en argument de la ligne de commande le paramètre *t* qui donne la durée en secondes de *sleep(t)*.

Dans un premier terminal, lancer une exécution avec une valeur *t=1000*. À l'aide d'un second terminal, commenter les données observables sur ce processus obtenues par la commande *ps*, et vérifiez que le processus se retrouve bien dans */proc*.

Créer un programme *fork\_arg\_sleep.c*, similaire à *fork.c* mais qui prend en paramètre deux variables donnant le temps d'endormissement du père et du fils respectivement.

Utiliser ce programme pour créer des processus zombies (vous pourrez voir leur présence à l'aide de la commande *ps -x* : les zombies sont représentés par l'état *Z*).

## 2 Exercice 2 : wait()

Programmer l'exemple du cours *exemple\_fork.c* (diapositive 9) et le tester.

Modifier le programme pour passer en paramètre deux chaînes de caractères qui seront affichées respectivement par le père (*argv[1]*) et le fils (*argv[2]*).

## 3 Exercice 3 : Boucle de création de processus

Modifier l'exemple du cours *n\_fork\_waitpid.c* (diapositive 15) pour :

- afficher les pid des processus qui terminent ;
- afficher aussi les codes de retour associés.

Un exemple d'exécution est :

```
$ ./n_fork_waitpid 5
```

Je suis le fils 0 (10359) du père 10358

Je suis le fils 1 (10360) du père 10358

Je suis le fils 2 (10361) du père 10358  
Le fils (10359) a terminé avec le code de retour 0  
Le fils (10360) a terminé avec le code de retour 1  
Le fils (10361) a terminé avec le code de retour 2  
Je suis le fils 4 (10363) du père 10358  
Je suis le fils 3 (10362) du père 10358  
Le fils (10363) a terminé avec le code de retour 4  
Le fils (10362) a terminé avec le code de retour 3