

TD1 – Processus (création et ordonnancement)

Objectifs

- Comprendre la création d'un processus avec `fork()`.
- Observer les **PID/PPID**.
- Manipuler les états d'un processus (running, stopped, zombie).
- Découvrir l'ordonnancement via `ps`, `top`, `kill`.

1. Rappel théorique

- `fork()` crée un nouveau processus (fils).
- Le fils hérite de la mémoire du père mais a son propre **PID**.
- `wait()` permet au père d'attendre son fils.
- États possibles : **R (running)**, **S (sleeping)**, **T (stopped)**, **Z (zombie)**.

2. Exercices

Exercice 1 : Créer un processus

Écrire un programme qui affiche :

- le PID du père,
- le PID du fils.

Attendu : deux lignes différentes avec `getpid()` et `getppid()`.

Exercice 2 : Zombie process

- Modifier pour que le fils termine **immédiatement**, mais que le père **ne fasse pas** `wait()`.
- Observer avec `ps` :

```
ps -o pid,ppid,stat,cmd | grep a.out
```

→ Le fils apparaît en état **Z**.

Exercice 3 : Processus suspendu et repris

- Créer un processus fils qui fait une boucle infinie.
- Observer avec top.
- Utiliser :

```
kill -STOP <PID>
```

```
kill -CONT <PID>
```

→ Vérifier le changement d'état (Running ↔ Stopped).

Exercice 4 : Ordonnancement

- Créer **deux fils**, chacun affichant son PID en boucle.
- Observer dans top comment le CPU alterne entre eux.
- Question : que se passe-t-il si on lance la commande nice ou renice ?

Commandes utiles

- ps -o pid,ppid,stat,cmd
- top ou htop
- kill -STOP, kill -CONT, kill -KILL
- nice, renice