

## TP n°5

### Les signaux

*Le but de ce TP est de manipuler les appels systèmes relatifs à la gestion des signaux.*

## 1 Positionnement de gestionnaire

Dans cet exercice, nous souhaitons vérifier le comportement de la fonction `sigaction`.

### Questions

1. Écrivez un programme qui se met en pause pendant 10 secondes à l'aide de la fonction `alarm`. Une fois les 10 secondes passées, le programme affiche un message puis termine "normalement".
2. Écrivez un programme qui se met en pause pendant 10 secondes et qui compte le nombre de signaux `SIGINT` reçus. Une fois la pause terminée, le programme affiche le nombre de signaux reçus.



Lorsqu'un gestionnaire est exécuté dans votre programme à la suite de la réception d'un signal, même si le programme doit s'arrêter, faites en sorte qu'il reprenne le cours normal d'exécution (dans le `main`, par exemple) avant de s'arrêter. Vous pouvez utiliser une variable globale `stop`, par exemple.

## 2 Prise en main des ensembles de signaux

L'objectif de cet exercice est de tester le fonctionnement de `sigprocmask`, `sigpending` et `sigaction`, ainsi que les fonctions de gestion des ensembles de signaux.

### Questions

1. Écrivez un programme qui bloque tous les signaux puis se met en pause pendant 20 secondes.
  - (a) Peut-on utiliser la fonction `sleep` pour la pause ?
  - (b) Et la fonction `alarm` ?
  - (c) Vérifiez que les signaux sont bien bloqués pendant la pause (augmentez la durée si nécessaire).
  - (d) Que se passe-t-il si le signal `SIGKILL` est envoyé ?
2. Après les 20 secondes de pause, le programme doit afficher les signaux reçus pendant sa pause.
  - (a) Modifiez le programme pour afficher le code des signaux reçus.
  - (b) Est-il possible de savoir si plusieurs signaux du même type ont été reçus ?

### 3 Signaux temps-réel

L'objectif de cet exercice est de manipuler les signaux temps-réel.

#### Questions

1. Dans cette question, nous souhaitons écrire deux programmes. Le premier programme attend la réception du signal `SIGRTMIN+1` et retourne ce même signal au processus qui lui a envoyé. Le deuxième programme prend en argument le PID du processus correspondant au premier programme et lui envoie un signal `SIGRTMIN+1`. Il se met ensuite en attente d'un signal `SIGRTMIN+1`.
  - (a) Comment le premier programme peut-il récupérer le PID du deuxième programme à la réception du signal ?
  - (b) Comment faire en sorte que l'attente soit une attente passive ?
  - (c) Écrivez les programmes.
2. Modifiez le deuxième programme pour qu'il envoie en plus un entier spécifié en argument (en plus du PID du premier programme). Le premier programme affiche l'entier reçu et renvoie le double. Le deuxième programme affiche la valeur de l'entier reçu.