

### **Exercices sur les signaux Linux**

Vos programmes et réponses aux questions seront à déposer sur [lecnam.net](http://lecnam.net) dans les dossiers de dépôt mis à votre disposition.

Le nom du programme déposé devra être sous la forme : votrenom\_exoN (avec N de 1 à 6).

Attention, dans le cadre du contrôle continu, certains des exercices vont être notés.

Enfin, dans la mesure du possible, il serait bien de proposer des procédures et/ou des fonctions pour les exercices les plus complexes.

#### **Exercice 1**

Reprendre l'exemple du cours et le faire fonctionner :

Créer un programme qui exécute une boucle infinie jusqu'à ce que le signal SIGHUP soit reçu par le processus.

#### **Exercice 2**

Écrire un programme qui puisse envoyer le signal SIGHUP au programme de l'exercice 1. On cherchera le PID du programme de l'exercice 1 et on lancera le programme de l'exercice 2 de la façon suivante : ./exo2 PID.

#### **Exercice 3**

Écrire une nouvelle version du programme 1 afin que ce dernier crée un fils qui exécute la boucle infinie. Le père se mettra en attente quelques secondes (commande sleep) puis enverra le signal SIGHUP au fils. Que constatez-vous ?

Puis faire l'inverse, c'est-à-dire que le père crée un fils puis exécute la boucle infinie et le fils se met en attente quelques secondes et envoie le signal SIGHUP. Que constatez-vous ?

#### **Exercice 4**

Écrire un programme qui redirige le traitement de SIGUSR1, affiche un message à chaque réception du signal par un autre processus et permet de s'arrêter en tapant CTRL C.

Modifier le programme afin d'afficher à chaque réception du signal le PID du processus distant qui envoie le signal.

Rajouter au programme la possibilité de rediriger le traitement de CTRL C pour écrire un message de fin.

#### **Exercice 5**

Écrire un programme qui affiche le numéro du signal reçu lorsqu'on tape CTRL-C et attend 10 secondes le temps de tester CTRL-C.

#### **Exercice 6**

Écrire un programme qui génère aléatoirement un nombre entier compris entre 1 et 1000 et permet à l'utilisateur de trouver ce nombre.

Ce dernier possède jusqu'à NB\_ESSAI\_MAX pour y arriver.

Sont comptabilisés comme un essai :

- 1) une tentative erronée
- 2) un temps de réponse trop long

Le point 2) devra être programmé à l'aide d'un signal. Attention, comment traite-t-on le signal et revient-on dans l'exécution en cours du programme ?