

## L3 Système TP3

### Exercice 1 - Highlander

Écrire un programme qui "résiste" au signal **SIGINT** (rappel: c'est ce signal qui est envoyé lorsque vous tapez Control-C dans le terminal). Pour cela utiliser l'appel système **signal()**.

### Exercice 2 - Highlander II

Réécrire le programme précédent avec l'appel système **sigaction()**. Pour manipuler des **sigset\_t**, consultez la manpage **sigsetops** (il faudra utiliser au moins **sigemptyset()**).

### Exercice 3 - speed-O-meter

Écrire un "speed-o-meter", un programme qui mesure la vitesse à laquelle on lui envoie des données (par l'entrée standard), en octets par seconde.

### Exercice 4 - shaper

Écrire un "shaper", un programme qui lit sur l'entrée standard et écrit sur la sortie standard, mais n'écrit pas plus vite qu'une certaine vitesse spécifiée en octets par seconde.

### Exercice 5 - Tentative de détournement, à mains nues

Écrire un programme qui redirige la sortie standard vers un fichier, puis effectue un "ls" vers ce fichier. Il faut donc ouvrir ce fichier avec **open()**, puis utiliser **dup2()** afin que les accès ultérieurs à la sortie standard se fassent sur le fichier. Ensuite, on peut faire un **execlp()** pour lancer **ls**, et l'affichage se fera vers le fichier.

### Exercice 6 - Tentative de détournement, à la fourchette

- Révisions sur **fork()**

Écrire un programme qui exécute le programme spécifié sur la ligne de commande, et en fonction de son code de retour, affiche "OK" ou "ERREUR".

Rappels : pour récupérer le code de retour, utiliser **waitpid()**; et OK correspond à un code de retour nul.

Exemple : programme [ "toto" = "titi" ] doit afficher ERREUR.

N'hésitez pas à consulter la manpage de la **commande /**, c'est instructif (**man bash**).

- Modifier le programme précédent pour qu'il n'affiche rien d'autre que OK ou ERREUR, c'est-à-dire que le programme qui sert de "condition" ne doit rien afficher.

Pour l'empêcher de faire de l'affichage, on propose de rediriger la sortie vers le fichier **/dev/null**.

Attention, il ne faut pas faire un simple **exec** mais un **fork + exec**.

Attention aussi à ne détourner la sortie standard que dans le fils, afin que le père puisse encore afficher OK ou ERREUR à la fin du programme.