

Exercice 1 : Lecture dans un tube sans rédacteur

Ecrire un programme qui tente de lire dans un tube sans rédacteur.

Exercice 2 : Ecriture dans un tube sans lecteur

Ecrire un programme qui tente d'écrire dans un tube sans lecteur. Quel signal est généré dans ce cas. Installer un handler qui intercepte ce signal.

Exercice 3 :

Lire l'aide pour consulter la syntaxe de la commande *kill* qui permet d'envoyer un signal à un processus et donc de l'arrêter puisque c'est le traitement par défaut associé.

Lancer un éditeur de textes en arrière plan, afficher son pid et utiliser la commande *kill* pour l'arrêter. Utiliser une option de la commande *kill* pour afficher tous les signaux du système.

Exercice 4 : Installation d'un handler

Reprendre le programme de la page 18 (voir support de cours). Vérifier qu'on ne peut pas arrêter ce programme avec le premier CTRL/C qui sera généré.

Exercice 5 :

Ecrire un programme qui ignore les signaux SIGINT et SIGQUIT pendant 30s et qui les rétablit après. **Indications** : Ignorer un signal consiste à installer le handler SIG_IGN pour ce signal.

Exercice 6 : Communication entre processus à l'aide d'un tube

Ecrire un programme qui crée un processus fils. Le père et le fils partagent un tube. Le père écrit dans l'ordre un opérateur, un opérande 1 et un opérande 2. Le fils devra lire ces informations pour effectuer l'opération (addition, multiplication, soustraction ou division) et afficher le résultat.

Les opérandes 1 et 2 sont des valeurs entières alors que l'opérateur correspond au caractère '-', '+', '*' ou '/'. Lorsque le père n'a plus rien à générer, il envoie le caractère '#' pour marquer la fin d'écriture dans le tube.