

TP N°2 système

Communication entre processus

I/ Signaux

Exercice n°1: Commande kill

Lancer un éditeur de texte en arrière plan, afficher son pid et utiliser la commande kill pour l'arrêter.

Exercice n°2 : Installation d'un handler

Ecrire un programme qui installe un handler pour le signal SIGINT avant de faire une boucle infinie. Le handler affichera un simple message. Vérifier qu'on ne peut pas arrêter ce programme avec Ctrl-C. Utiliser d'autres touches pour arrêter ce programme.

Indications: - La primitive `signal(num_signal, fonction)` permet d'installer le handler `fonction` pour le signal `num_signal`. Inclure `<signal.h>`

- Pour arrêter ce programme, on peut utiliser Ctrl-\. Ces touches peuvent changer d'un clavier à un autre. Vous trouverez la combinaison des touches qu'il faut utiliser à l'aide de la commande `stty -a`

Exercice n°3 :Handler SIG_DFL

Modifier le programme précédent pour qu'il soit arrêté au 10^{ème} Ctrl/C tapé. Pour cela il faudra installer au départ un nouveau handler qui compte le nombre de Ctrl/C tapés et au bout du 9^{ème} Ctrl/C installe le handler par défaut.

Indication: Installer le handler par défaut consiste à remplacer `fonction` de la primitive `signal(num_signal, fonction)` par `SIG_DFL`

Exercice n°4 :

- Ecrire un programme qui se duplique
- Le fils envoie un signal SIGUSR1 à son père toutes les secondes
- Le père compte les signaux reçus (SIGUSR1) Jusqu'à ce que l'utilisateur tape Ctrl-C

II/ Mémoire mappée

Le service système `mmap()` du noyau Unix permet d'associer des zones de l'espace d'adresse d'un processus à une zone d'un fichier

II.1 Lecture d'un fichier

En utilisant le service système `mmap` vu en cours, écrire un programme qui lit le contenu d'un fichier et l'écrit sur la sortie standard. On utilisera le service système `fstat()` pour déterminer la taille de la zone mémoire à associer.

II.2 Partage d'une zone mémoire

Le pseudo-périphérique `/dev/zero` réagit à une association par `mmap()` par la création d'une zone de mémoire physique, accessible sous ce nom par les processus fils de celui qui a fait le `mmap()`. En utilisant `mmap()`, `fork()` et `/dev/zero`, écrire un programme qui crée une association avec `/dev/zero` et ensuite crée un fils, qui doit lire la zone et y écrire les 1000 premiers entiers, que le père doit lire et afficher sur la sortie standard.