

## TP 3 : Communication inter-processus sous UNIX

### Les signaux

#### Capture des signaux

Les signaux sont des moyens de notification d'événements (interruptions logicielles). Ils ont pour but d'informer l'arrivée des événements. Exemple : SIGINT (2), SIGQUIT (3), SIGKILL (9).

#### Syntaxe

**Headers : sys/types.h, signal.h**

**int kill(pid\_t pid, int sig) /\* L'envoi du signal sig au processus pid**

A l'arrivée d'un signal le système exécute le gestionnaire par défaut de ce dernier. Le système peut exécuter un autre gestionnaire au lieu du gestionnaire par défaut. Ce mécanisme s'appelle la capture des signaux.

#### Syntaxe

**Header : signal.h**

**sighandler\_t signal(int signum, sighandler\_t handler) /\* Capturer le signal signum et exécuter la procédure handler à la place du gestionnaire par défaut**

**Exercice :** Ecrire un code C dans lequel un processus père crée un seul processus fils. Le père envoie deux signaux **SIGINT** et **SIGQUIT** à son fils. Pour éviter sa suspension et sa terminaison, le fils se protégera de ces signaux en les capturant avec un simple affichage d'un message de réception du signal.

Modifier le code pour capturer cette fois-ci le signal **SIGKILL** que le père enverra à son fils vers la fin d'exécution. Quelle conclusion peut-on tirer ?

**PS :** Afin d'être sûr que le processus fils reçoive tous les signaux avant sa terminaison, ralentissez son exécution et celle du père avec l'appel système **sleep(s)** permettant d'endormir un processus pendant s microsecondes.