

## TP N°3-a : Processus & signaux – 2exercices – Durée approx. ~ 1h00min.

A l'aide du support de cours et des éventuels mémentos réalisez les exercices suivants :

*(Tout code ou implémentation compilant ou non sera étudié)*

*Le présent sujet de TP comporte 2 pages*

### **I – « Compteur »**

Écrivez un programme comptant le nombre de signal de type **SIGINT** reçu.

- Le processus se compose d'une boucle sans fin qui incrémente un compteur associé au signal **SIGINT** : lorsque l'utilisateur appuie sur **CTRL+C**, le programme affiche le nombre sur la sortie standard.
- A la réception du signal **SIGTERM** le programme affiche un message de fin puis se termine.

### **II – Ping-Pong**

Le but de ce programme est de simuler un jeu de ping-pong (via les signaux **SIGUSR1** & **SIGUSR2**) en processus père et son fils. Pour ce faire, réalisez l'exercice suivant en réalisant les étapes dans l'ordre spécifié :

1. Écrivez un programme qui crée un processus fils.
2. Faites envoyer un signal **SIGUSR1** vers le processus fils.
3. Associez une fonction *sig\_handler* au signal affichant un message sur la sortie standard STDOUT : « signal reçu ».
4. Faites évoluer cette fonction de manière à ce que cette dernière renvoie un signal **SIGUSR1** vers le processus père via la commande **signal**.
5. Idem pour le processus père.
6. Rajouter une temporisation au sein des fonctions afin que les signaux soient temporisés.
7. Ajouter une notion d'aléatoire symbolisant le fait que le joueur rattrape ou non la balle.  
*(ie : 0 -> le joueur manque la balle ; 1 -> il la renvoie)*
8. Comptez le nombre de points : le premier à 13 affiche un message sur la sortie standard : « j'ai gagné ».
9. Le gagnant se charge d'envoyer le signal **SIGTERM** au processus adversaire.

## **IV – Rappels**

### **Approche incrémentale du développement**

Pour obtenir les résultats attendus aux différents exercices, veuillez toujours appliquer une approche incrémentale en termes d'ajout de code/fonctionnalité.

Par exemple: une approche incrémentale pour ce type d'exercice "Is-like" serait :

1. La récupération des paramètres.
2. Tester fichier/répertoire.
3. Parcourir les éléments du répertoire en affichant leur nom.
4. Alimenter chaque fichier avec une information supplémentaire : permission / taille / propriétaire...

### **Documentation**

Pour obtenir des informations ou de la documentation ayez le réflexe d'utiliser les pages du manuel.

Par exemple:

➤ man 3 stat / man 2 open / man 2 readdir / man errno

### **Gestion des erreurs**

Afin d'avoir une gestion des erreurs la plus précise possible ayez le réflexe d'utiliser les codes retours **ERRNO** spécifiés dans les pages de manuel

Par exemple:

- |                  |   |
|------------------|---|
| ➤ <b>EEXIST</b>  | File exists (POSIX.1)                       |
| ➤ <b>EFAULT</b>  | Bad address (POSIX.1)                       |
| ➤ <b>EISDIR</b>  | Is a directory (POSIX.1)                    |
| ➤ <b>ENOTDIR</b> | Not a directory (POSIX.1)                   |
| ➤ <b>ELOOP</b>   | Too many levels of symbolic links (POSIX.1) |