

Feuille de TD4 : Signaux et tubes

Rappels :

```
kill(pid, signal); // envoie le signal « signal » au processus identifié par pid
if (signal(s,fonction)==SIG_ERR) // si le programme reçoit le signal "s" alors la
fonction est lancée.
```

```
void fonction(int s) {
printf("signal %d reçu \n",s);
}
```

NSIG nombre de signaux sur votre machine.

Exercice 1 : Proposer un programme *c* qui attrape tous les signaux. Dès qu'il attrape un signal, il affiche le signal reçu.

Utiliser la commande *kill* dans le terminal sous le format suivant : *kill -signal pid*. (Essayez les commandes *ctrl+c*, *ctrl+z*, de redimensionner la fenêtre...)

Exercice 2 : Proposer un programme *c* qui demande un pid (entier) et envoie tous les signaux au premier programme.

Exercice 3 : Génocide des fils

Donner un programme *c* qui génère *n* fils (*n* est passé en argument à la commande). Chaque fils écrit son PID ainsi que son numéro de création toutes les secondes. Le père tue tous ses fils après 5 secondes. (Signal 9)

Exercice 4 : ping pong avec tubes non nommés

```
int tube[2]; //stock le descripteur de fichier.
pipe(tube); //crée le tube anonyme.
close(tube[0]); // ferme le fichier.
read/write //....
```

Donner un programme *c* qui crée 2 fils. Quand le premier fils reçoit le signal 3 alors il envoie le signal 4 au second fils (en affichant « ping »). Le second fils envoie le signal 3 quand il reçoit le signal 4 (en affichant « pong »). Le père commence la partie de ping pong en envoyant le signal 3 au premier fils puis il arrête la partie au bout de 10 secondes.

Exercice 5 : discussion entre père et fils (tube)

Donner un programme *c* qui crée un processus. Le père demande un texte dans le terminal, l'envoie au fils et le fils l'affiche dans le terminal (en indiquant qu'il est le fils).

Exercice 6 : discussion entre père et fils avec réponse (Double tubes)

Donner un programme *c* qui crée un processus. Le père demande un texte dans le terminal, l'envoie au fils et le fils affiche qu'il l'a bien reçu puis le renvoie au père qui l'affichera.

Exercice 7 : discussion entre frères

Donner un programme *c* qui crée un processus. Le premier fils demande un texte dans le terminal, l'envoie au second fils et le second fils affiche qu'il l'a bien reçu puis le renvoie au premier fils qui l'affichera.

Exercice 8 : discussion entre le père et ses fils avec réponse (Plusieurs tubes)

Donner un programme *c* qui crée *n* processus. Le père demande un texte dans le terminal et l'envoie à tous ces fils puis chaque fils affiche le message en indiquant qui il est.

Exercice 9: Utilisation d'un tube nommé de communication

Faire deux programmes le premier demande un message sur le terminal et écrit le message dans le tube le second attend le message du tube pour l'afficher.

```
if (mkfifo("fichier.tube", 0644)!=0)// création du fichier pour l'utiliser en TUBE
{perror("mkfifo");exit(1);}//Si création du tube échouée
```

Les deux programmes doivent être utilisés dans deux terminaux différents.

Exercice 10 : Discussion

Les deux programmes précédents doivent poursuivre la discussion. C'est-à-dire le second programme demande un message sur le terminal et l'envoie au premier et ainsi de suite.