

But du TP

Le but de ce TP est de mettre en avant la création de processus, la communication entre processus par les signaux et les traitements particuliers qui peuvent être associés aux signaux.

Réalisation de la commande. Réaliser une commande **pan** intégrant les différents exercices sous forme d'options comme indiqué par l'exemple ci-dessous (ceci de manière à simplifier la correction des TP). Un appel de la commande sans argument provoque l'affichage des options :

```
% pan
--ex1 : Infanticide
--ex2 : Agonie
--ex3 : Debut de resistance
--ex4 : Duel
```

Dans chaque exercice, deux processus vont devoir afficher des messages, ces messages devront être préfixés par **Pere** ou **Fils** de manière à distinguer le processus réalisant l'affichage.

Exercice 1 : Préparation

Préparer la structure du programme. Écrire la fonction **main** qui, en fonction de l'option de la ligne de commande appelle la fonction correspondante. Écrire une fonction **aide**, à appeler lorsqu'aucune option n'est présente, qui affiche le détail des options.

Exercice 2 : Infanticide

Écrire l'option **--ex1**. Le processus doit créer un fils, ce fils affiche simplement un message annonçant sa création et son **pid** puis attend 10s et affiche un message annonçant sa terminaison normale. Dans un premier temps, le père affiche simplement son **pid** ainsi que celui de son fils. Tester le programme pour en vérifier le fonctionnement.

Exemple d'exécution.

```
% pan --ex1
Pere >> Pere de pid 1360 creation d'un fils de pid 1361
Fils >> Fils de pid 1361
% Fils >> Terminaison normale du fils
```

Modifier votre programme pour que le père envoie un signal de terminaison au fils (autrement dit, le père tue le fils). Il affiche "Pan" si l'envoi du signal s'est déroulé sans problème et un message d'erreur dans le cas contraire.

Exemple d'exécution.

```
% pan --ex1
Fils >> Fils de pid 1490
Pere >> Pere de pid 1489 creation d'un fils de pid 1490
Pere >> Pan !
```

Le message **Terminaison normale du fils** ne s'affiche plus, c'est normal, le fils est "tué" avant d'avoir eu le temps de réaliser l'affichage.

Il peut parfois être utile d'utiliser une attente (**sleep(n)**) pour que tous les processus puissent réaliser leur phase d'initialisation avant de commencer l'envoi des signaux.

Exercice 3 : Agonie

Ajouter l'option **--ex2**. Reprendre l'exercice précédent en modifiant le comportement du fils pour qu'il affiche le message "Argh, je meurs..." avant de mourir. Le père doit quant à lui se réjouir de cette nouvelle et afficher le message "Niak niak niak j'ai eu 104 !" si le **pid** du fils est 104.

Exemple d'exécution.

```
% pan --ex2
Pere >> Pere de pid 10249 creation d'un fils de pid 10250
Fils >> Fils de pid 10250
Pere >> Pan !
Fils >> Argh, je meurs...
Pere >> Niak niak niak j'ai eu 10250 !
```

Utiliser la primitive `wait` pour prendre connaissance de la terminaison du fils.

Exercice 4 : Début de résistance

Reprendre l'exercice précédent en ajoutant de nouvelles règles :

- le fils possède au départ un nombre de points de vie entre 1 et 10 (tiré au hasard) inconnu par le père (afficher ce nombre à l'écran) ;
- lorsque le fils possède un nombre de points de vie supérieur à 1 et qu'il est touché, il affiche "**Aie !**" et perd un point de vie ;
- lorsqu'il ne reste qu'un point de vie au fils et qu'il est touché, il affiche "**Argh, je meurs...**" et se termine ;
- le père continue de tirer jusqu'à ce que le fils meure, il manifeste alors son contentement par le message "**Niak niak niak !**"

Évidemment, il s'agit d'un TP sur les signaux, il est donc recommandé de choisir une solution basée sur les signaux. Après une phase d'initialisation, le corps du programme a une forme :

```
while (1) {
    sleep(1);
}
```

Exemple d'exécution.

```
%pan --ex3
Pere >> Pere de pid 10261 creation d'un fils de pid 10262
Fils >> Fils de pid 10262 nb points = 2
Pere >> Pan !
Fils >> Aie !
Pere >> Pan !
Fils >> Argh, je meurs...
Pere >> Niak niak niak !
```

Utiliser la primitive `waitpid` en mode non bloquant pour détecter la mort (comme la survie) du fils.

Exercice 5 : Duel

Dans cet exercice, les chances sont enfin égales, le père et le fils possèdent chacun un nombre de points de vie (entre 1 et 10) tiré au hasard et inconnu de l'adversaire. Les deux processus se lancent des signaux de terminaison en affichant le message "**Pan !**" tant que l'adversaire résiste. Lorsqu'un processus reçoit un signal de terminaison, soit il meurt en affichant le message "**Argh, je meurs...**" et en envoyant un signal d'agonie à son adversaire, soit il affiche le message "**Même pas mal !**" et le signale à son adversaire.

Exemple d'exécution.

```
%pan --ex4
Pere >> Pere de pid 13206 creation d'un fils de pid 13207
Fils >> Fils de pid 13207
Fils >> nb points = 6
Pere >> nb points = 3
Fils >> Pan !
Pere >> Meme pas mal !
Pere >> Pan !
Fils >> Grrr ! Pan !
Fils >> Meme pas mal !
Fils >> Grrr ! Pan !
Pere >> Grrr ! Pan !
Pere >> Meme pas mal !
Pere >> Grrr ! Pan !
Fils >> Grrr ! Pan !
Fils >> Meme pas mal !
Fils >> Grrr ! Pan !
Pere >> Grrr ! Pan !
Pere >> Argh, je meurs...
Fils >> Niak niak niak
```