Université de Rouen
I.U.P. 2
Module : Système I

T.P. 7 : Les signaux

La manipulation des signaux peut s'effectuer avec les appels système de la bibliothèque standard C.

Envoie d'un signal

```
#include <signal.h>
int kill(int pid, int sig);
```

La fonction `kill` envoie le signal de numéro `sig` au processus défini par `pid`.

Modification du handler

```
void (*signal(int sig, void (*handler()))();
```

Cette fonction fixe le comportement du processus lorsqu'il prendra en compte l'envoi d'un exemplaire du signal `sig`. La fonction `handler` sera alors exécutée.

Attente d'un signal

```
int pause(void);
```

La fonction fait passer le processus appelant à l'état endormi sur l'évènement "arrivée d'un signal quelconque". Le processus est reveillé par l'arrivée d'un signal quelconque.

Exercice 1 Ecrire un programme qui pendant un temps t donné, compte le nombre de `<INTR>` et `<QUIT>` frappé au clavier et qui affiche ces nombres avant de se terminer.

Exercice 2 Ecrire une fonction utilisant l'appel-système `alarm` et ayant le même fonctionnement que la fonction `sleep`.

Exercice 3 Un processus se duplique. Le père doit rester vivant tant que le fils est vivant. Si le fils se termine, le père doit se terminer aussitot. Un `<CTRL C>` envoyé au père doit arrêter les deux processus.

Exercice 4 Un processus crée un fils. Le père ignore les interruptions mais pas le fils. Lorsque le fils se termine, le père n'ignore plus les interruptions.

Exercice 5 On veut créer un protocole d'échange de signaux entre deux processus. Un père et son fils s'échangent 50 signaux `SIGUSR` avant de se terminer. Le fils émet un signal vers son père et se met en attente de l'accusé de reception. Ecrire le programme avec les primitives du TD. Expliquer pourquoi ce programme peut ne pas fonctionner correctement. Quel mécanisme supplémentaire doit-on posseder pour réaliser ceci ?