TP n°8 : Communication et synchronisation (les signaux)

Objectif : Envoi et réception de signaux entre processus

Travail à effectuer :

Le but de cet exercice est de mettre en pratique le mécanisme d'envoi de signaux entre processus dans Unix. Reprendre le cours et regarder dans le manuel les spécifications des fonctions `signal()` et `kill()` de manipulation des signaux.

1) Ecrire le programme ci-dessous et faites le tourner. Le signal *SIGINT* correspond à l'interruption du terminal (*CTRL-C*). Son action par défaut est la terminaison.

```
#include <stdio.h>
#include <signal.h>

void handINT (int signo){
      signal (SIGINT, handINT) ;
      printf ("reception d'un signal SIGINT\n");
}

void main(){
      signal (SIGINT, handINT) ;
      printf("processus : %d\n", getpid());
      while (1) ;
}
```

2) Ecrire un programme C dont le comportement est une boucle infinie et qui écrit à l'écran « *BONJOUR* » lorsqu'il reçoit le signal *SIGUSR1* et « *BONSOIR* » lorsqu'il reçoit le signal *SIGUSR2*. Ce programme ne doit réagir qu'une seule fois au signal *SIGUSR1*. Par contre il doit réagir autant de fois que le signal *SIGUSR2* lui arrive.

Indice : Regarder la fonction `pause()` dans le manuel. Attention l'action par défaut des signaux *SIGUSR1* et *SIGUSR2* est la terminaison.

3) Ecrire un programme dans lequel un processus crée un fils et initialise un handler (afficher « *BONJOUR* ») sur *SIGUSR1*. Le fils affiche des informations à l'écran puis envoie le signal *SIGUSR1* à son père. Attention le programme fils doit se terminer avant le processus père.