TD6 : Processus (2)

Exercice 1 : Environnement d'un processus

La variable globale **environ** est déclarée comme :

```
extern char **environ
```

Cette variable est automatiquement définie. Elle référence le premier élément d'un tableau de chaînes de caractères de la forme *var=valeur*. Ce tableau est terminé par le pointeur NULL.

1. Dessiner cette structure de données
2. Ecrire la fonction de librairie **getenv** qui permet de récupérer la valeur d'une variable du shell. Le résultat est le contenu de la variable *var*, ou 0 si elle n'existe pas.

Exercice 2 : Signaux

Ecrire un programme composé d'une boucle sans fin qui incrémente un compteur. Lorsque l'utilisateur appuie sur la touche d'interruption (signal SIGINT), le programme sauvegarde la date en clair et la valeur courante du compteur dans un fichier, à la suite de ce qui s'y trouve déjà. Lorsque le signal SIGTERM est reçu, le programme écrit le mot *fin* dans le fichier, puis se termine.

Indications : Vous devez utiliserez les primitives POSIX suivantes :

- La primitive **sigaction()** qui permet d'installer la fonction associée à un signal. Elle a comme argument la structure **sigaction** qui est composée des champs **sa_handler** (pointeur vers la fonction) et **sa_flags** (options).
- La primitive **sigemptyset ()** qui initialise l'ensemble des signaux masqués (**action.sa_mask**).
- La primitive **sigaddset ()** qui permet d'ajouter un signal à l'ensemble des signaux masqués (**action.sa_mask**).

Exercice 3 : Signaux

A l'aide de la fonction **kill()** et des signaux SIGSTOP et SIGCONT, écrire le programme où :

1. Un processus père crée un processus fils qui compte de 1 à 5 (affichage par seconde). Trois seconde après sa création, le processus fils est mis en pause par le processus père, qui le relancera après avoir attendu cinq secondes.
2. Que se passe-t-il si le signal SIGINT est envoyé au lieu de SIGSTOP ?