

## TME 5 / 6 : GESTION DE PROCESSUS

### ÉCRITURE D'UNE VERSION PARALLELE DE LA COMMANDE GREP

**Objectif** : Manipulation des processus sous Unix.

**Utilisation des appels systèmes** : `fork`, `execl`, `wait` et `wait3`.

On souhaite implanter en C un "multi-grep" qui exécute la commande standard Unix `grep` en parallèle. `grep` permet de rechercher une chaîne de caractères dans un fichier et elle affiche les lignes où la chaîne apparaît.

L'exécution de votre programme devra être appelée de la manière suivante:

```
$ mgrep chaine liste-fichiers
```

Cette commande devra afficher, pour chaque fichier passé en paramètre, les lignes contenant la chaîne `chaine`. Elle créera autant de processus fils que de fichiers passés en paramètre.

#### 1. MULTI-GREP SIMPLE

Ecrivez un programme C qui lance, pour chaque fichier passé en paramètre, un processus fils qui exécute le `grep` standard. Le programme (c'est-à-dire le père) ne doit se terminer que lorsque tous les fils ont terminé.

#### 2. MULTI-GREP A PARALLELISME CONTRAINT

On souhaite désormais ne créer simultanément qu'un nombre maximum `MAXFILS` de processus fils. Si le nombre de fichiers est supérieur à `MAXFILS`, le processus père ne crée dans un premier temps que `MAXFILS` fils. Dès qu'un des fils se termine et s'il reste des fichiers à analyser, le père recrée un nouveau fils.

##### 2.1

Modifiez votre programme en conséquence

#### 3. MULTI-GREP AVEC STATISTIQUES

---

**3.1**

Modifiez votre programme de la partie 1 pour afficher les statistiques d'utilisation CPU système et utilisateur de chaque fils.

<b>4. PROCESSUS "ZOMBIE"</b>
------------------------------

---

**4.1**

Qu'est ce qu'un processus "zombie" ?

---

**4.2**

Ecrivez un programme qui crée pendant 10 secondes un processus zombie.