

# Contrôle R3-05

## Programmation système

### Exercice 1 : Threads et semaphore

Nous souhaitons écrire un programme multithread permettant à l'utilisateur de :

1. créer un tableau **V** ordonnée d'entiers de taille **N** définie par ce dernier. Ce tableau contient des entiers allant d'une valeur initiale **V0** fixée par l'utilisateur et les entiers générés sont par conséquent compris entre **V0** et **V0+N-1**. Par exemple si **N=5** et **V0=120**, le tableau **V** contiendra donc la suite suivante : **V={120, 121, 122, 123, 124}**
2. écrire les valeurs du tableau dans un ordre décroissant dans un fichier dont le nom est saisi par l'utilisateur
3. L'utilisateur vérifie le contenu de son fichier en affichant les valeurs à la console

Pour cela vous allez respecter les exigences suivantes :

- Tous les paramètres choisis par l'utilisateurs doivent être des paramètres positionnels
- Votre programme doit contenir cette fonction dont la signature est comme suit :
  - `Int tableau2fichier(int * tableau, int taille)` qui permet de :
    - Créer N threads  $T_i$  où chaque  $T_i$  écrit la valeur de la case n°  $i$  du tableau V pour tout  $0 \leq i \leq N-1$
    - D'utiliser la structure de données suivante
      - `typedef struct structThread {  
    int idThread ;  
    int val ;  
} std ;`
- La fonction du thread doit se nommer **jobThread**
- Votre programme doit contenir une fonction qui se nomme **creerTableau** permettant de créer le tableau d'entier dans l'ordre croissant entre **V0** et **V0+N-1**, dont la signature est comme suit : `int * creerTableau(int V0, int N)`
- Votre main doit
  - appeler `creerTableau`
  - appeler `tableau2fichier`
  - Afficher le contenu du fichier une fois tous les threads sont terminés

**Question à rendre sur papier :**

**-écrire l'algorithme de la fonction `jobThread` en précisant la zone critique.**

**Pour la suite vous pouvez choisir entre l'exercice 2 et l'exercice 3**

### Exercice 2 : Communication par Tubes

On souhaite écrire un programme permettant au processus principal **P0** de créer deux processus fils **F1** et **F2** où **F1** et **F2** doivent lire via des tubes une même chaîne de caractères nommée **mot** :

- **F1** doit lire la chaîne **mot** via son tube, calcule la somme **Sentier** des entiers contenus dans **mot** et envoie via un tube la valeur **Sentier** calculée
- **F2** doit lire la chaîne **mot** via son tube, calcule la somme **ScoreAsc** des codes ascii des caractères (qui ne sont pas des chiffres) contenus dans la chaîne et envoie **ScoreAsc** au processus père via un tube

- *P0 doit récupérer les deux sommes et les affiche en précisant le PID système du processus fils qui a envoyé la valeur.*

### **Exercice 3 : Processus**

*Ecrire un programme qui permet à un processus père P0*

- *de récupérer une chaîne de caractères définie en paramètre positionnel,*
- *crée N processus fils où N est la taille de la chaîne de caractères*
- *chaque processus fils doit écrire dans un fichier partagé par tous les processus, nommé partage.txt, sur une ligne son PID Système et la chaîne de caractères*
- *le processus principal P0 doit attendre la fin de tous les processus fils pour créer un nouveau processus fils dont le rôle est de lire le fichier partage.txt, affiche à la console son contenu*
- *le processus principal P0 doit attendre la fin de ce dernier pour supprimer le fichier partage.txt.*