

Module de « Systèmes d'exploitation II » : Série d'exercices de Travaux Pratiques

Exercice 1 : Mise en scelle.

- Installer le package JDK de Sun sur Ubuntu.
- Ecrire le programme Java suivant qui affiche « Bonjour » sur l'écran. Le compiler et l'exécuter.

```
/ Ma premiere application en Java
class FirstApp {
    public static void main (String[] args){
        System.out.println("Bonjour");
    }
}
```

Exercice 2 : Goûter aux threads !

On veut écrire un programme Java qui crée deux threads. Chacun des deux threads affiche les nombres entiers de 1 à N (N est choisi par l'utilisateur au moment de l'exécution).

Ecrire le programme précédent en deux versions en mettant en œuvre les deux méthodes de création de threads vues en cours.

Exercice 3 : Jongler avec les threads.

On considère 3 types de threads : Carré, Rectangle, Triangle .

Un thread affiche en permanence la figure géométrique qu'il désigne (carré ou rectangle ou triangle) en mouvement de rotation.

Faire un programme Java qui crée, à la demande, un thread quelconque (Carré, Rectangle, Triangle). Le programme doit permettre d'interrompre, de reprendre et de finir les threads.

TP final : Une seule variante à traiter (voir l'affectation faite par binôme)

Sujet : Trois types de processus accèdent à une liste partagée :

- ceux qui font de la recherche (S)
 - ceux qui font l'insertion (I)
 - ceux qui font l'effacement (D)
-
- Les processus S examinent la liste, donc ils peuvent le faire de façon concurrente entre eux.
 - Les processus I ajoutent des nouvelles cellules à la fin de la liste, donc ils doivent être en exclusion mutuelle entre eux; toutefois, un processus I peut agir en concurrence avec un nombre quelconque de processus de type S.
 - Les processus D effacent des cellules à tout endroit de la liste, donc ils doivent agir en exclusion mutuelle entre eux et avec les I et S.

Ce problème peut être résolu en plusieurs variantes, en fonction de la priorité qu'on décide de donner à chaque type de processus :

- Variante 1 : Priorité(S)>Priorité(D)>Priorité(I)
- Variante 2 : Priorité(S)>Priorité(I)>Priorité(D)
- Variante 3 : Priorité(D)> Priorité(S)>Priorité(I)
- Variante 4 : Priorité(D)> Priorité(I)>Priorité(S)
- Variante 5 : Priorité(I)> Priorité(D)>Priorité(S)
- Variante 6 : Priorité(I)> Priorité(S)>Priorité(D)

Travail à faire : Ecrire un programme solution en java de la variante choisie. Le programme doit permettre de :

- Créer et de terminer interactivement des threads (S, I, D).
- De visualiser (graphiquement) l'évolution des tous les threads.

Le programme doit être accompagné d'un rapport + disquette qui contiennent : Description des fichiers (classes) utilisés, Code source en java.

Evaluation : Le travail sera évalué sur la base des critères suivants : Démonstration du programme (10 points), Rapport (05 points), Réponses aux questions (05 points).

Echéance : Remise du rapport et démonstration sur machine : Janvier 2011.

NB : Les travaux non personnels seront sanctionnés par un zéro.