

Université de Bretagne Sud  
UFR SSI, Dept MIS  
CONCURRENCE  
TP : Les Threads JAVA

Luc Courtrai

## 1 Serveur de tâches

On désire faire exécuter à un programme des tâches en parallèle. Pour simplifier l'exercice chaque tâche aura à imprimer une suite de caractères sur la sortie standard à un intervalle de temps. Chaque tâche doit être prise en charge par un threads Java.

La liste des tâches est décrite dans un fichier d'ordres dont voici un exemple (*donnees.jobs*) :

```
a 10 2
b 20 1
c 15 3
```

Dans cet exemple, la première tâche imprime 10 caractères *a* sur la sortie standard. Entre chaque impression de caractère, 2 secondes doivent s'écouler.

Le programme *ParCar* lit les données sur l'entrée standard :

```
java ParCar donnee.jobs
```

Ecrire en Java le programme *ParCar*.

Exemple de lecture des lignes d'un fichier :

```
import java.io.*;
..
try {
    // Le fichier d'entrée
    File file = new File("donnees.jobs");
    // Créer l'objet File Reader
    FileReader fr = new FileReader(file);
    // Créer l'objet BufferedReader
    BufferedReader br = new BufferedReader(fr);

    String line;
    while((line = br.readLine()) != null) {
        System.out.println(line);
        // scinde la ligne en mot
        String mot[] = ligne.split(" ");
    }
} catch(IOException e) {
    e.printStackTrace();
}
```

## 2 find

Écrire une version parallèle de la commande Find où le traitement de chaque répertoire sera pris en charge par un nouveau thread Java.

La commande affiche au fur et à mesure les répertoires dans lesquels se trouve l'entrée recherchée.

On limitera les arguments et les fonctionnalités du find au cas suivant

usage : java Find nomRecherche repertoire

Avec le nomRecherché exact de l'entrée (pas d'expression régulière)

Pour vous aider voici en java d'un ls

```
File repertoire =new File(args[0]);

if (repertoire.isDirectory()) {
    File[] listeFichiers = repertoire.listFiles();

    LinkedList<Thread> works= new LinkedList<Thread>();
    for( File entree : listeFichiers){
        System.out.println(entree.getName());
    }
}
```

### not found

Essayer d'ajouter le message "not found" si la recherche échoue. Il faut pour cela, outre l'ajout d'une variable static, attendre (join) la fin des tous les threads créés.