Java Avancé

Cours 5 : Programmation Parallèle et Asynchrone

Arsène Lapostolet

2 Février 2024

Introduction

Thread

Thread : fil d'exécution. Plusieurs en même temps possible.

- Améliorer la vitesse d'un process : attention à l'overhead
- Permettre un cas d'utilisation : ex modèle client-serveur

Thread Java

- Classe `Thread`, construit à partir d'un `Runnable`
- Lancer avec la méthode `start()`

```
var oddThread = new Thread(() -> {
    IntStream
            .range(0, 100)
            .filter(number -> number % 2 != 0)
            .forEach(System.out::println);
});
```

```
var evenThread = new Thread(() -> {
    IntStream
            .range(0, 100)
            .filter(number -> number % 2 == 0)
            .forEach(System.out::println);
});

evenThread.start();
oddThread.start();
```

```
Thread.sleep(5000);
```

Synchronisation

Joindre

Ressource partagée et section critique

Synchroniser

Programmation Asynchrone

Définition

Notion d'E/S non bloquantes

Asynchrone en Java

Thread Pool :

```
ExecutorService threadPool = Executors.newFixedThreadPool(4);
```

CompletableFuture :

```
CompletableFuture<String> task = CompletableFuture.supplyAsync(()
-> {
    var result = ... opération longue à exécuter de façon
asynchrone...
```

```
    return result;
});

task.thenAccept((String result) -> System.out.println(result));

task.exceptionally(exception -> {
   exception.printStackTrace();
   return "";
});
```