



CentraleSupélec

Option OSY
Programmation avancée

TP 2

Multithreading

Exercice 1 : Echauffement

Version naïve

Créez une classe possédant un champ public statique/global *Compteur* de type entier initialisé à 0. Créez 10000 threads dont le but est d'incrémenter et décrémenter 10000 fois le champ *Compteur*, puis de l'incrémenter une dernière fois.

Que remarquez-vous ?

Version améliorée 1

Mettez en place une synchronisation permettant d'attendre la fin de l'exécution des 10000 threads. Que remarquez-vous ?

Version améliorée 2

Synchronisez les threads pour l'accès à *Compteur*.

Exercice 2 : Spooler

Nous allons simuler à présent le fonctionnement d'un périphérique du genre imprimante réseau. Pour cela on va simuler des documents de plusieurs pages ; 1 document sera 1 ligne de la forme :

Thread <#id> : page <#page>

Où <#id> est le numéro du thread et #page est le numéro de la page. Le nombre de pages est choisi au hasard entre 1 et 10. L'impression d'une page dure 500ms.

Version naïve

Implémentez une première version du programme sans synchronisation. Le logiciel lance une dizaine de threads qui affichent les lignes sur la console. Que remarquez-vous ?

Version améliorée 1

Synchronisez les threads pour que les impressions ne soient pas interrompues.

Version améliorée 2

On a une imprimante qui peut imprimer deux documents à la fois (du coup, à l'affichage, on peut mélanger les pages de deux documents). Modifiez le code précédent pour prendre en compte cela.

Exercice 3 : Producteur / consommateur

Dans cet exercice nous allons modéliser un problème dans lequel nous avons deux threads:

- Le premier thread produit un résultat et met un certain temps à le calculer (un temps aléatoire entre 1 et 3 secondes) : le résultat est mis dans une file d'attente. En pratique, ce thread va créer les entiers de 1 à 10.

- Le second thread consomme les éléments dans la file d'attente (et met entre 1 et 3 secondes): le résultat est alors affiché.

Ecrire un programme qui modélise un tel système.