

PROJET SD 2013-2014

ORDONNANCEUR DE TÂCHES JAVA

Modalités

Ce projet est à réaliser en binôme. Vous déposerez sur ARCHE une archive (.zip ou .tar.gz) contenant le code source de votre projet ainsi qu'un rapport synthétique du projet (5 pages maximum, format pdf) détaillant la conception de votre application (descriptions des classes sous la forme de diagrammes UML), les difficultés rencontrées et comment vous les avez résolues, ainsi que le nombre d'heures passées sur les différentes étapes (conception, codage, tests, rédaction du rapport) par chaque membre du groupe.

Le dépôt des projets se fera sur la plateforme ARCHE à l'adresse :

<http://arche.univ-lorraine.fr/mod/assign/view.php?id=190711>

avant le 9 mai 2014 à 23h55

(aucune soumission en retard ne sera acceptée)

Comme pour le module POO, un projet ne compilant pas sera sanctionné par une note adéquate. Des soutenances individuelles de projet seront organisées à une date ultérieure. Vous serez jugé sur la qualité de votre programme, celle de votre rapport et votre capacité à expliquer son fonctionnement.

Vous devrez mettre en œuvre les connaissances acquises (héritage, interfaces, structures de données, algorithmes, gestion des flux) lors des différents modules de programmation (notamment TOP, POO et SD). Le choix des structures de données est libre, mais doit être pertinent et justifié dans le rapport.

Ne trichez pas ! Ne copiez pas ! Si vous le faites, vous serez lourdement sanctionnés. Nous ne ferons pas de distinction entre copieur et copié. Vous n'avez pas de (bonne) raison de copier.

Présentation du problème

L'objectif de ce projet est de concevoir, développer et tester un ordonnanceur de tâches en Java. Cet ordonnanceur prendra en entrée un

fichier contenant une liste de tâches à exécuter (chacune de ces tâches étant caractérisée par un numéro, un intitulé, un instant d'arrivée, une durée, et éventuellement une priorité (valeur entière)), sous la forme suivante :

Politique FIFO				
Numéro	Intitulé	Arrivée	Durée	Priorité
1	Informatique	0	7	-
2	Courses	1	3	-
3	Sport	2	2	-
4	Informatique	2	10	-
5	Maths	1	3	-

Il fournira en retour un diagramme d'ordonnement sous la forme :

1-Informatique (0 -> 7)
2-Courses (7 -> 10)
5-Maths (10 -> 13)
3-Sport (13->15)
4-Informatique (15->25)

Des statistiques seront également affichées concernant le temps moyen de séjour, le temps moyen d'attente, le nombre de changements de contexte.

Ce diagramme et les statistiques pourront être affichés à l'écran ou être stocké dans un fichier (option `-o NomFichier`).

Plusieurs politiques d'ordonnement seront implantées :

- Politique **FIFO** ou **First In First Out** : les tâches sont traitées par ordre d'arrivée,
- Politique **SJF** ou **Shortest Job First** : les tâches les plus courtes sont traitées prioritairement,
- Politique **RR(q)** ou **Round Robin(q)** : elle est également appelée politique du tourniquet. Les tâches sont traitées alternativement selon un quantum de temps **q** donné,

- Politique **PR ou Priority** : chaque tâche est associée à une valeur de priorité (plus la valeur est haute, plus la tâche est prioritaire).

Lorsque deux tâches de même priorité sont en concurrence, elles seront départagées en fonction de leur numéro (numéro le plus faible étant le plus prioritaire dans ce cas).

Plus d'informations sur les algorithmes d'ordonnement sont disponibles à l'adresse suivante (chapitre sur l'ord. des processus) : <http://www.groupe.polymtl.ca/inf2610/documentation/notes/chap8.pdf>

Vous étendrez ensuite votre ordonnanceur à deux politiques avec préemption (où une tâche peut être arrêtée pour laisser place à une autre tâche), à savoir :

- Politique **PFIFO ou Preemptive First In First Out** : les tâches sont affectées d'une valeur de priorité et peuvent être remises en attente à l'arrivée d'une tâche plus prioritaire,
- Politique **SRT (Shortest Remaining Time)** : cela correspond à la version préemptive de la politique SJF. L'arrivée d'une tâche plus courte peut provoquer un changement de contexte.

Vous implanterez également un **mode interactif** (option `-i`). Dans ce cas, l'application ne prendra pas de fichier en entrée, mais l'utilisateur saisira les données au fur et à mesure : incrémentation du temps, déclaration de l'arrivée d'une nouvelle tâche. Des résultats partiels de l'ordonnement pourront être affichés à chaque nouvel événement saisi par l'utilisateur.

Bonus (tâches facultatives)

Les tâches suivantes peuvent également être considérées :

- Génération des fichiers résultats au format JSON (JavaScript Object Notation, en utilisant une bibliothèque dédiée), cela correspondra à l'option `-json`,
- Ordonnement multi-cœurs où plusieurs tâches peuvent être traitées en parallèle (option `-coeur NbDeCoeurs`),
- Extension à d'autres politiques d'ordonnement de votre choix,
- Interface graphique avec la bibliothèque Swing.