Ordonnancement de tâches

Eric Goubault et Matthieu Martel

Le but de ce TD est d'implémenter un ordonnanceur de tâches dépendantes les unes des autres. Les dépendances, représentées par un DAG, seront données en entrée et l'on supposera que l'on dispose d'un nombre borné p de processeurs.

Question 1 Dans un premier temps, on écrira un ordonnanceur générique de liste, selon le schéma suivant :

1. Initialisation:
   - Calculer le niveau de priorité de toutes les tâches,
   - Constituer la file de priorité comme étant la liste des tâches libres (sans prédécesseur) triées en ordre de priorité décroissante.
2. Tant qu'il reste des tâches à exécuter:
   - Ajouter les nouvelles tâches libres à la file de priorité. Si l'exécution d'une tâche se termine, supprimer cette tâche de la liste des prédécesseurs de chacun de ses successeurs,
   - Ajouter les tâches dont la liste des prédécesseurs est devenue vide à la file,
   - S'il y a p processeurs disponibles et r tâches dans la file, supprimer les premières tâches et les exécuter.

Les différentes classes qu'il sera nécessaire d'implémenter sont:

- Une classe pour les tâches à exécuter, contenant des champs indiquant leur priorité, leur poids, ainsi que les dépendances.
- Une classe "Liste d'attente" permettant de gérer les tâches en attente d'exécution.
- Une classe pour les processeurs qui prennent la première tâche de la liste et l'exécute.
- Une classe pour l'ordonnanceur, se chargeant d'ajouter les nouvelles tâches dans la liste d'attente lorsque celles-ci sont prêtes (les tâches dont elles dépendent ont terminé leur exécution).

Les synchronisations entre la liste d'attente, l'ordonnanceur et les processeurs peuvent être gérées de deux façons: soit à l'aide de synchronisation (wait/notify), soit en modifiant les priorités des tâches comme nous l'avons vu pour l'ordonnancement par tourniquet.

Question 2 Modifier le point 1 de la phase d'initialisation du programme précédent afin que les priorités affectées aux tâches correspondent à un ordonnancement basé sur le chemin critique. Il sera ici nécessaire d'implémenter une classe permettant de gérer les DAGs.