

## Contrôle Continu Théorie et Pratique de la Concurrency

Jeudi 7 Avril 2016. Durée : 2 heures.

Document autorisé : Une feuille A4 recto-verso manuscrite

### Exercice 1

*Construire de l'eau*

Il existe deux types de threads, **Oxygène** et **Hydrogène**. Pour assembler ces threads dans des molécules d'eau ( $H_2O$ ), on va créer une *barrière* qui est attendue par chaque thread tant qu'il n'y a pas suffisamment de threads pour créer une molécule d'eau (composée de deux threads **Hydrogène** et un thread **Oxygène**). Juste avant qu'un thread passe la barrière, il va afficher un message. Il faut garantir que les messages des threads qui forment une molécule sont affichés avant les messages des threads d'une autre molécule. Plus précisément, le programme devra respecter les conditions suivantes :

- si un thread **Oxygène** arrive à la barrière quand aucun thread **Hydrogène** n'est présent, il doit attendre deux threads **Hydrogène**,
- si un thread **Hydrogène** arrive à la barrière quand aucun autre thread n'est présent, il doit attendre un thread **Oxygène** et un autre thread **Hydrogène**.

Les threads doivent passer la barrière dans des ensembles complets ; ainsi, en examinant la séquence de messages affichés et en les divisant en groupe de trois, chaque groupe doit contenir un thread **Oxygène** et deux threads **Hydrogène**.

On souhaite développer un programme concurrent simulant le comportement décrit ci-dessus :

1. Définir une méthode `waitBarrier()` qui bloque jusqu'au moment où trois threads rentrent dans cette méthode. En plus, la méthode doit garantir que les threads sortent de cette méthode par groupe de trois. Cette méthode ne prendra pas en compte le fait que les threads soient des threads **Hydrogène** ou **Oxygène** et devra utiliser la mémoire partagée.
2. Écrire un programme concurrent qui modélise le comportement des atomes **Oxygène** et **Hydrogène** décrit ci-dessus. Ce programme pourra utiliser la méthode `waitBarrier()`.
3. Décrire les modifications que vous devez apporter à votre solution pour traiter le cas de l'eau oxygénée  $H_2O_2$  (deux atomes **Oxygène** et deux atomes **Hydrogène**).

Donnez un programme (en Java ou en C ou en pseudo-code comme en cours) pour **chaque** de ces trois questions. Le mode de communication des processus au sein de vos programmes se fera par mémoire partagée. Pour vos programmes, si vous utilisez des sémaphores, vous pourrez supposer qu'il s'agit de sémaphores forts. Pour la réponse, merci de préciser à chaque fois la question à laquelle correspond votre programme. Vos programmes devront en plus garantir l'absence d'interblocage.