HMIN105M - TP noté 1

Durée : 1h15 (**à déposer impérativement avant 16h15)**

23 octobre 2018

Dans un jeu, N joueurs doivent suivre un parcours constitué de 3 étapes. Chaque étape ne peut supporter qu'un nombre limité de joueurs simultanément ($1 \leq$ nombre de joueurs d'une étape $\leq N$). Dépassé cette limite, un joueur doit attendre qu'un autre joueur avance dans le jeu avant de pouvoir progresser à son tour. Tous les joueurs doivent aller jusqu'au bout du jeu.

Exemple : soit étape1, étape2 et étape3 les étapes du jeu et NbJoeursEtape1 le nombre maximum de joueurs à pouvoir passer cette étape en même temps (NbJoeursEtape2 pour étape2 et NbJoeursEtape3 pour étape3). Le schéma algorithmique global d'un joueur sera :

Joueur :
si moins de NbJoeursEtape1 sont à l'étape 1, progresser, sinon attendre ;
étape1 ;
si moins de NbJoeursEtape2 sont à l'étape 2, progresser, sinon attendre ;
étape2 ;
si moins de NbJoeursEtape3 sont à l'étape 3, progresser, sinon attendre ;
étape3 ;
arrivée à la fin du parcours.

Pour mettre en oeuvre la possibilité d'avoir un ou plusieurs joueurs à réaliser une même étape, on définit une structure de synchronisation, appelée *N-verrou*. Cette structure permet à k ($1 \leq k \leq N$) threads d'être présents simultanément en section critique, mais pas plus de k threads à la fois. Pour utiliser des *N-verrou*, on dispose des 4 fonctions suivantes :

- **int n_verrou_init(n_verrou * v, int k)** pour initialiser la structure n_verrou. k est le nombre maximum de threads pouvant être simultanément en section critique.
- **int n_verrou_lock(n_verrou * v)** pour demander l'entrée dans la section critique.
- **int n_verrou_unlock(n_verrou * v)** pour avertir de la sortie de la section critique.
- **int n_verrou_destroy(n_verrou * v)** pour détruire les éléments de la structure n_verrou.

L'ensemble de ces fonctions renvoie 0 en cas de succès, -1 en cas d'échec.

- Télécharger les fichiers fournis. Vous y trouverez un squelette de code à étudier et à compléter (jeu.c) ainsi que la réalisation d'une étape (ne pas ré-implémenter) et un Makefile pour compiler. La fonction réalisant un joueur est fournie et ne doit pas être modifiée.
- Compléter la définition des structures utilisées dans le code.
- Réaliser une implémentation des fonctions précédentes.
- Ecrire un programme mettant en oeuvre le jeu de l'énoncé. Attention : le nombre total de joueurs et le nombre maximum de joueurs pour chaque étapes doivent être des paramètres de votre programme (voir le code).
- Testez votre programme avec différents paramètres. Vous remarquerez que la durée d'une étape varie entre 3 et 15 secondes.

Dépôt de votre travail Vous devez déposer un seul fichier contenant tout votre code source (le fichier jeu.c). Tout fichier supplémentaire sera automatiquement refusé. Votre dépôt doit être anonyme : votre nom/prénom/pseudo ne doit pas figurer ni dans le nom de fichier, ni dans son contenu, ni dans la description du dépôt. Lors du dépôt, dans la case commentaire/description, dites ce qui fonctionne et/ou ne fonctionne pas.