HMIN101M - TP noté 1

Durée : 1h15

10 octobre 2017

REMARQUE : Lisez attentivement l'énoncé avant de commencer le travail.

On souhaite exécuter en parallèle (par plusieurs threads) une séquence de traitements, tout en contrôlant le nombre maximum de threads pouvant exécuter simultanément un traitement de cette séquence.

Exemple : supposons que 10 threads sont lancés pour exécuter chacun la séquence de traitements suivante :

```
Role d'un thread :
        traitement1() ;
        traitement2() ;
        traitement3() ;
```

et que traitement2() est un traitement qui ne peut être exécuté simultanément par plus de 3 threads (exemple pouvant nécessiter un tel comportement : un accès partagé au réseau pour transférer de grands volumes de données tout en assurant un débit de transmission satisfaisant).

L'objectif est de lancer les 10 threads pour pouvoir exécuter les traitements des différents threads en parallèle, mais d'empêcher un thread de commencer traitement2() si 3 threads sont déjà en cours d'exécution de traitement2(). Lorsqu'un thread termine traitement2(), un autre thread qui est en attente peut poursuivre son exécution. Seul traitement2() est concerné par cette contrainte.

Pour mettre en oeuvre un tel comportement, on défini une structure de synchronisation, appelée *N-verrou*, qui permet à k ($1 \leq k \leq N$) threads d'être présents simultanément en section critique, mais pas plus de N threads à la fois. Pour utiliser des *N-verrou*, on souhaite disposer 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 signaler 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 renvoi 0 en cas de succès, -1 en cas d'échec.

- Définir la structure n_verrou,
- Réalisez une implémentation des fonctions précédentes.
- Ecrire un programme mettant en oeuvre l'exemple de l'énoncé. Attention : le nombre total de threads et le nombre maximum de threads doivent être des paramètres de votre programme (ou des données à saisir au clavier). Les fonctions traitement1(), traitement2() et traitement3() se contenteront d'endormir le thread appelant de quelques secondes (sleep(..)). Veillez à ce que le temps de traitement3() soit supérieur à celui de traitement2().

Remarque Votre travail sera évalué en grande partie via son exécution et sans avoir lu le code. L'exécution de votre programme doit donc clairement illustrer l'exécution parallèle des threads et la résolution du problème décrit dans ce sujet.

Dépôt de votre travail Vous devez déposer un seul fichier contenant tout votre code source (un fichier .c ou .cpp. 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, fournissez la ligne de commande pour la compilation de votre code.