Cours de Systèmes Programmés Enfouis
Examen de Première Session 2008

Livres fermés. Durée : 3 h 30

Répondez à chaque question sur une feuille séparée sur laquelle figurent nom, prénom et section.

1. Une carte électronique chargée de piloter un écran LCD est commandée via un bus I^2C. Elle reçoit des ordres sous forme de paquets, séparés d'au moins 100 μs.

   Cette carte doit exécuter les tâches suivantes:

   - Une tâche τ_1 chargée de réceptionner les paquets de données reçus et de les enregistrer en mémoire interne.
   - Une fois qu'un nombre suffisant de paquets a été reçu, la tâche τ_2 effectue un contrôle d'intégrité sur ceux-ci, ainsi qu'un traitement permettant de les transformer en un message compréhensible par la tâche τ_3. Un mécanisme de file d'attente est implémenté pour pouvoir accumuler des messages.
   - Une tâche τ_3 chargée de contrôler l'écran en interprétant le message en tête de la file d'attente gérée par τ_2. L'opération peut être complexe (dessins de figures géométriques, de caractères, ...). Le contrôle de l'écran se fait par l'écriture à un endroit particulier de la mémoire de données.

   Le temps d'exécution de τ_1 est beaucoup plus court que l'intervalle de temps séparant l'arrivée de deux paquets, ce qui garantit le bon fonctionnement du protocole. Le temps d'exécution de τ_2 est de l'ordre de 80 μs. Le temps d'exécution de τ_3 peut, dans le pire des cas, aller jusqu'à 500 ms.

   Remarques: Les paquets venant du bus I^2C sont signalés par une interruption. Si le test d'intégrité de la tâche τ_2 ne réussit pas, les paquets concernés sont ignorés. Il se peut que la file gérée par la tâche τ_2 soit pleine lorsqu'un message doit être ajouté. Dans ce cas, le message en tête de la file d'attente est supprimé, et le message récent est ajouté.

   (a) Quel schéma d'organisation interne du logiciel enfoui est le plus approprié au problème posé? Justifiez.

(b) Donnez, au moyen de pseudocode, la structure globale du logiciel de la carte.

2. (a) Expliquez ce qu'est le problème d'inversion de priorité.
   (b) Le problème d'inversion de priorité peut-il se manifester dans le cas de deux tâches? Justifiez.
   (c) Qu'est-ce que la stratégie d'ordonnancement à taux monotone?
   (d) Si n tâches périodiques ordonnançables utilisent pleinement le processeur, peut-on, en général, ajouter au système des tâches qui n'ont pas de contraintes particulières? Justifiez.

3. Dans le cadre du traitement d'un signal, on désire vérifier le bon fonctionnement d'un analyseur de spectre. Celui-ci emploie un micro-contrôleur qui reçoit en entrée un signal analogique et l'échantillonne grâce à un convertisseur analogique/digital. Dès la mise en marche du système, le micro-contrôleur effectue des mesures à la fréquence de 2048 Hz et les place dans un *buffer* circulaire de taille 2080. L'analyseur de spectre doit effectuer périodiquement une transformée de Fourier rapide (*Fast Fourier Transform*, FFT) appliquée sur les 2048 derniers échantillons. Initialement, le micro-contrôleur attend donc d'avoir collecté 2048 échantillons avant d'exécuter la première FFT.

   Pendant que la FFT est exécutée, le système continue à échantillonner le signal.

   Voici quelques précisions:

   - Le temps de calcul de la FFT est compris entre $\frac{1}{70}$ et $\frac{1}{60}$ seconde.
   - Le calcul de la FFT s'effectue toutes les $\frac{1}{32}$ secondes (c'est-à-dire exactement le temps requis pour collecter 64 mesures).

   (a) Modélisez l'analyseur de spectre par un système hybride. A tout moment, une variable doit retenir l'indice du *buffer* dans lequel l'échantillon suivant sera placé. Aussi, pendant le calcul de la FFT, une variable doit retenir l'indice du *buffer* où débutent les 2048 échantillons traités.
   (b) On veut vérifier que le système n'introduit pas de nouveaux échantillons dans la partie de *buffer* traitée par la FFT durant son exécution. Comment peut-on le faire en se servant du système hybride?

   Remarques: Le contenu du *buffer* et les valeurs du signal n'ont pas besoin d'être modélisés. On ne demande pas d'effectuer une exploration de l'espace des états accessibles du système.

Bon travail!