

FONDEMENTS DES SYSTEMES EMBARQUES TEMPS-REEL

SPECIALITE SAR, UNIVERSITE PARIS VI

barème indicatif, **sans document**

Les téléphones portables doivent être éteints et rangés dans vos sacs.

Les parties de l'examen sont indépendantes.

Il sera tenu compte de la présentation et de la clarté dans la rédaction.

Seules les réponses précises et justifiées seront considérées

1 Ordonnancement temps réel (6pts)

On vous demande d'assister un ingénieur qui doit modéliser et analyser une application embarquée dans un véhicule automobile. Le système à analyser est constitué d'un processeur qui héberge les fonctionnalités suivantes :

1. Analyse de l'état du véhicule grâce à des capteurs afin d'assurer la sécurité des passagers.
2. Contrôle du moteur. En particulier, en cas de danger, le processeur diminue la vitesse, voire immobilise automatiquement le véhicule.
3. Rafraîchissement du tableau de bord qui donne au conducteur l'accès aux informations produites par les deux fonctionnalités ci-dessus.

Ces trois fonctionnalités sont implantées par trois tâches indépendantes sur un système d'exploitation temps réel. Le système d'exploitation offre un service d'ordonnancement basé sur un ordonnancement à priorité fixe préemptif dont les niveaux de priorité vont de 0 à 31 (0 est le niveau de priorité le plus élevé).

Vous proposez à l'ingénieur de modéliser ces tâches comme des tâches périodiques avec les paramètres suivants :

1. La tâche T1 (analyse de l'état du véhicule) est une tâche périodique dont la période est de 10 millisecondes. Son temps d'exécution est de moins d'une milliseconde.
2. La tâche T2 (contrôle du moteur) est une tâche périodique dont la période est de 15 millisecondes et son temps d'exécution est de 5 millisecondes exactement.
3. La tâche T3 (rafraîchissement du tableau de bord) est une tâche périodique dont la période est de 60 millisecondes et son temps d'exécution est de 20 millisecondes.

Toutes les tâches arrivent dans le système à l'instant zéro et les délais critiques sont égaux aux périodes.

1.1 Configuration du scenario.

L'ordonnanceur étant à priorité fixe, il est nécessaire d'assigner une priorité à chaque tâche.

Généralement, selon quels critères assigne-t-on les priorités fixes ?

Proposez une priorité pour les trois tâches ci-dessus selon l'algorithme Rate Monotonic.

1.2 Chronogramme du scenario.

Ordonnancer ces tâches en mode préemptif sur l'hyper-période.

1.3 Analyse des temps de réponse.

Déterminer le pire temps de réponse des tâches à partir du chronogramme.

1.4 *Nouveau scenario.*

Finalement, l'ingénieur vous indique que les trois tâches accèdent à une zone mémoire commune. Cette zone mémoire est protégée par une section critique. Pourquoi les temps de réponse déterminés dans la question 3 sont-ils faux ? Comment contourne-t-on en général le problème pour se ramener au cas précédent en restant dans un cadre d'ordonnancement préemptif ?

Veillez répondre à cette partie sur une feuille séparée

2 Langage Java (3 pts)

- Après avoir rappelé leurs caractéristiques, donnez un scénario dans lequel un `NoHeapRealTimeThread` et un `RealTimeThread` peuvent être impliqués dans un cas d'inversion de priorité.
- Expliquez le fonctionnement de la méthode `waitForNextPeriod` en l'illustrant par un exemple (les fautes de syntaxe ne sont pas importantes).
- Quelle est la différence entre une zone mémoire de type `LTMemory` et celle de type `VTMemory` ?

3 Bus et réseaux (3 pts)

- Dans le cas de l'ordonnancement EDF, donner, en la commentant, la formule qui permet de modifier la date de réveil d'une tâche T de façon à prendre en compte les dépendances entre tâches communicant par messages. **Rappeler les hypothèses.**
- Où s'arrête l'analogie entre l'ordonnancement des tâches sur un processeur et celui des messages sur un réseau ? Comment exprime-t-on ceci dans la formule d'ordonnançabilité de tâches RMS.
- Soit 3 sites, S1, S2 et S3 interconnectés par un bus de type CAN, leurs priorités sont telles que : $\text{Priorité}(S1) > \text{Priorité}(S2) > \text{Priorité}(S3)$. Le numéro de priorité de S3 est 10.
Comment numéroter les priorités de S1 et S2 de façon à ce que ces sites prennent le bus en fonction de leur priorité ? Décrire précisément l'accès au bus.

4 Noyaux temps réel – POSIX (3 pts)

- L'édition de liens et la pagination peuvent chacun introduire un facteur d'indéterminisme. Dans chaque cas :
 1. expliquer précisément pourquoi,
 2. indiquer comment remédier au problème.
- Quel dysfonctionnement peut intervenir lors de l'utilisation d'un `pthread_cond_wait` qui doit être débloqué par un `pthread_cond_broadcast` ?
- Expliquer précisément comment réaliser une attente temporisée en utilisant une variable conditionnelle.

5 Tolérance aux fautes (5pts)

5.1 *Traitement des erreurs.*

Les deux affirmations suivantes concernent une application logicielle s'exécutant sur un réseau de machines interconnectées par un réseau sûr de fonctionnement. Dites si l'affirmation est vraie ou fausse en justifiant votre choix.

- 1) Les fautes du logiciel peuvent être tolérées grâce à l'utilisation de la diversification des implémentations (1 service – N implémentations) tant que les erreurs qu'elles engendrent ne se propagent pas au matériel.
- 2) La réplication passive augmente la disponibilité d'un système, et la réplication active sa fiabilité.

5.2 *Analyse d'un exemple.*

Nous nous intéressons au logiciel permettant de réguler la vitesse d'une voiture autour d'une valeur d'équilibre choisie par le conducteur (description simplifiée non réaliste).

Lorsqu'il est en fonction, le régulateur doit maintenir la vitesse courante malgré les variations de la pente de la route (cause principale de variation de vitesse). La voiture est conçue de telle sorte que l'accélération du moteur est commandée numériquement à travers la lecture périodique d'une variable partagée entre le contrôle moteur et le régulateur de vitesse.

La vitesse de la voiture est transmise au régulateur qui met à jour la valeur de l'accélération pour maintenir la vitesse autour de la valeur souhaitée. Toutefois, la valeur proposée par le régulateur est soumise à une vérification avant écriture dans la mémoire partagée : l'écart entre deux valeurs consécutives doit rester inférieur à une valeur prédéfinie A .

Soit a_1 la dernière valeur proposée et a_2 la valeur recommandée par le régulateur, alors la valeur inscrite dans la mémoire partagée est :

$$a_3 = a_1 + \text{signe}(a_2 - a_1) * \max(A, |a_2 - a_1|)$$

Avec *signe* la fonction qui retourne le signe d'un nombre (positif si nul), *max* la fonction maximum, et $|\cdot|$ la fonction qui retourne la valeur absolue d'un nombre.

- 1) Définissez dans un cadre général les notions de *sécurité-innocuité* et *composant auto-testable*.
- 2) Sur l'exemple :
 - a. Proposez un scénario d'exécution du système permettant d'illustrer la mise en défaut d'une propriété de sécurité-innocuité par rapport au conducteur ?
 - b. En quoi peut-on considérer que le régulateur de vitesse est un composant auto-testable ?
- 3) Définissez en général les notions de réplifications active et passive. Dites pourquoi la réplification active est applicable ici pour assurer la tolérance aux fautes du régulateur. (indice : nature de l'exécution)