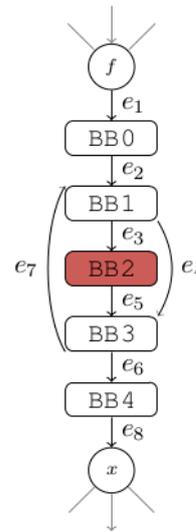


COMASIC – SETI 2017
Contrôle de Connaissances
UE de Systèmes Temps Réel Embarqués Critiques
17 novembre 2017
2h00 - Sans document – Barème indicatif

1 Pire temps d'exécution (3 pts)

On considère le code et le graphe de flot de contrôle ci-dessous pour les questions suivantes:

```
#define MAXI 10
void f() {
  // BB0
  int i = 0;
  while (i < MAXI) {
    // BB1
    if (i == 0) {
      // BB2
    }
    // BB3
  }
  // BB4
  i++;
}
```



1.1 Analyse statique du pire temps d'exécution [cours] (1 point)

Quelles sont les étapes principales de l'analyse statique du pire temps d'exécution en considérant l'approche IPET (Implicit Path Enumeration Technique) ? Expliquez quel type d'information est calculé durant chaque étape. Expliquez également comment ces informations sont utilisées durant les étapes suivantes. Décrivez comment ces informations sont intégrées dans les équations IPET. Il n'est pas nécessaire de détailler les formules mathématiques.

1.2 Flot du programme sous IPET (1 point)

Donnez les équations linéaires d'IPET pour la fonction f ci-dessus. Vous pouvez utiliser les variables f et x afin de représenter le flot entrant/sortant de la fonction. Utilisez les variables des arcs du graphe de flot de contrôle (e_n) dans les équations. Il n'est pas nécessaire (pour l'instant) de donner des équations concernant des bornes de la boucle ("flow facts"), ni la fonction objectif. Expliquez brièvement chaque équation.

1.3 Flot du programme avancé (1 point)

Donnez des équations supplémentaires concernant le flot dans la boucle ainsi que le flot du bloc de base [BB2]. Expliquez brièvement chaque équation.

Veillez répondre aux parties suivantes sur une feuille séparée

2 Comparaison d'ordonnancements [cours] (2 points)

Rappeler les principes de fonctionnement de EDF et LLF. Comparer ces ordonnancements dans un contexte mono-cœurs. De même, comparer leurs versions globales pour multi-cœurs.

3 Serveur périodique de tâches apériodiques [cours] (2 points)

Rappeler le fonctionnement d'un serveur sporadique. Décrire son fonctionnement sur un exemple de votre choix. Indiquer les variantes éventuelles. La qualité de la rédaction et des explications sera notée.

4 Bus CAN [cours] (2 points)

Pourquoi le protocole d'accès au bus CAN est-il qualifié de *bitwise arbitration* (CSMA/BA) ? Justifier sur un exemple mettant en jeu trois sites. Expliquer comment ce bus permet de modéliser les messages envoyés sous forme de tâches exécutées et préciser l'algorithme d'ordonnement sur lequel il se fonde.

5 Integrated Modular Avionic (6 points)

On souhaite exécuter un ensemble de tâches sur une plate-forme partitionnée de type ARINC 653. L'ensemble de tâches se compose de :

Nom	Capacité	Période/Echéance
T1	1	3
T2	1	6
T3	2	6
T4	2	12

Les échéances sont égales aux périodes. La date de première activation de toutes les tâches est identique et est égale à zéro. Cependant, T1 démarre légèrement avant T2, de même que T2 avec T3 et T3 avec T4. On souhaite appliquer un ordonnancement Rate Monotonic préemptif. L'arbitrage de l'ordonnancement en présence de tâches de priorité identique est laissé à votre discrétion.

5.1 Première table d'ordonnement fondée sur RMS (2 points)

On cherche à ordonner ce jeu de tâches sur une architecture comportant un seul processeur. Expliquer pourquoi le test d'ordonnancement ne permet pas de conclure sur l'ordonnancement de ce lot de tâches. Etablir un possible chronogramme montrant l'exécution de ce lot de tâches sur l'hyper-période.

5.2 Définition de la criticité [cours] (2 points)

On établit que les tâches T1 et T3 ont même niveau de criticité A et T2 et T4 ont même niveau de criticité B. Rappeler la signification du niveau de criticité en termes de probabilité de défaillance dans le contexte avionique. Rappeler également les couvertures de code attendues pour les niveaux de criticité.

5.3 Partitionnement d'une application critique (2 points)

On souhaite rassembler les tâches par niveau de criticité pour former des partitions. En s'appuyant sur l'ordonnement précédent et en supposant que l'ordonnement des partitions reste Rate Monotonic, énumérer les fenêtres de partition (*partition windows*) que cela nécessite ainsi que le nombre de changements de partition. Vous minimiserez le nombre de changements de partition grâce à l'arbitrage de tâches de priorité identique.

Veillez répondre aux parties suivantes sur une feuille séparée

6 RT-POSIX (5 pts)

Soit 1 mutex M et 3 threads POSIX T1, T2 et T3 dont le code est donné ci-dessous. T1, T2 et T3 sont périodiques de périodes respectives 50 ms, 100 ms et 200 ms. La politique d'ordonnancement considérée est RMS. Les priorités POSIX des tâches T1, T2 et T3 sont respectivement: 15, 10, et 5. Les threads sont configurées avec la politique SCHED_FIFO. Enfin, on considère les temps d'exécution suivants :

- Fonc1_part1 a un temps d'exécution de 5 ms ;
- Fonc1_part2 a un temps d'exécution de 5 ms ;
- Fonc1_part3 a un temps d'exécution de 5 ms ;
- Fonc2 a un temps d'exécution de 20 ms ;
- Fonc3_part1 a un temps d'exécution de 40 ms ;
- Fonc3_part2 a un temps d'exécution de 15 ms ;
- Fonc3_part3 a un temps d'exécution de 20 ms ;

T1	T2	T3
Fonc1_part1() ; pthread_mutex_lock(M) ; Fonc1_part2() ; pthread_mutex_unlock(M) ; Fonc1_part3() ;	Fonc2();	Fonc3_part1() ; pthread_mutex_lock(M) ; Fonc3_part2() ; pthread_mutex_unlock(M) ; Fonc3_part3() ;

On néglige le temps d'acquisition et de relâche du verrou. Nous donnons ci-dessous le code d'initialisation du mutex M:

```
pthread_mutex_t M;
pthread_mutexattr_t mutex_attr;
pthread_mutexattr_setprotocol(&mutex_attr, PTHREAD_PRIO_PROTECT);
pthread_mutexattr_setprioceiling(&mutex_attr, 15);
pthread_mutex_init(&M, &mutex_attr)
```

6.1 Définition de (Immediate) Priority Ceiling Protocol [cours](1.5 points)

Rappeler le fonctionnement de Priority Ceiling Protocol. Rappeler ce qui le différencie de Immediate Priority Ceiling Protocol.

6.2 Utilisation d'un protocole de synchronisation (2 points)

En prenant en compte l'initialisation du mutex M, représentez le chronogramme d'exécution du lot de tâches proposé.

6.3 Calcul du pire temps de blocage (1.5 points)

Calculez le pire temps de blocage dû à l'utilisation de M, pour les tâches T1, T2, et T3.