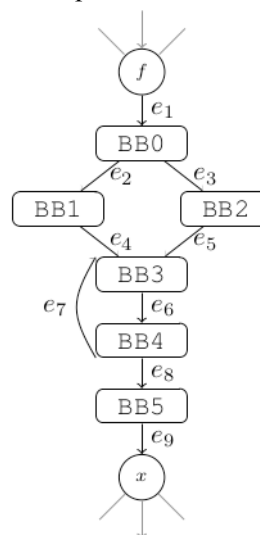


ASTRE - SE301 2017
Contrôle de Connaissances
UE de Systèmes Temps Réel Embarqués Critiques
23 novembre 2017
2h00 - Sans document – Barème indicatif

1 Pire temps d'exécution (4 points)

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

```
void foo(bool large_block, int *b)
{
  // BB0
  if (large_block) n = 10; // BB1
  else n = 5; // BB2
  while (n) { // BB3
    // BB4
    n--;
  } // BB5
}
```



1.1 Le pire-temps d'exécution [cours] (2 points)

Donnez une définition du terme borne de pire-temps d'exécution (Worst-Case Execution Time Bound). Quelles caractéristiques du programme et/ou du matériel peuvent avoir un impacte sur les bornes obtenues ? Quelles sont les caractéristiques souhaitées et les difficultés associées avec le calcul de ces bornes ?

1.2 Flot du programme sous IPET (1 point)

Donnez les équations linéaires d'IPET pour la fonction `foo` 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, 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 en considérant les deux bornes $n = 5$ et $n = 10$ venant des blocs de bases `BB1` et `BB2`.

VEUILLEZ REpondre AUX PARTIES SUIVANTES SUR DES FEUILLES SEPARÉES

2 ARINC 653 (4 pts)

2.1 Communication inter-partition [cours] (1 point)

En ARINC653, quels sont les mécanismes de communication inter-partition ? A quel(s) moment(s) les transferts de données entre partitions sont-ils effectués ?

2.2 Communication inter-partition (3 points)

Donner la configuration temporelle (i.e. MAF, MIF, partition windows) d'un système ARINC653 à partir des informations contenues dans le tableau suivant.

Tâche	Période	Capacité (temps d'exécution)	Priorité	Criticité
T1	50	10	4	A
T2	100	20	2	A
T3	200	40	3	B
T4	25	5	5	C
T5	100	20	1	C

VEUILLEZ REpondre AUX PARTIES SUIVANTES SUR DES FEUILLES SEPAREES

3 Ordonnancement Multi-Coeurs (5 points)

Soit le jeu de tâches suivant constitué de 4 tâches à échéances implicites (i.e les échéances sont égales aux périodes) dont les paramètres de temps d'exécution pire cas (C_i) et de périodes (T_i) sont :

$$\tau_1 = (C_1, T_1) = (4, 6) ; \tau_2 = (C_2, T_2) = (7, 12) ; \tau_3 = (C_3, T_3) = (4, 12) ; \tau_4 = (C_4, T_4) = (10, 24)$$

Nous souhaitons ordonnancer ce jeu de tâches sur deux processeurs P1 et P2. Pour représenter un ordonnancement de ce jeu de tâches, voici un modèle de chronogramme à reporter sur votre copie. Pour représenter l'exécution d'une tâche sur un processeur dessiner un rectangle de la longueur du temps d'exécution correspondant et noter à l'intérieur de ce rectangle soit P1 soit P2 pour indiquer quel processeur est utilisé.

T1																				
T2																				
T3																				
T4																				

En cas d'égalité sur les valeurs de priorités entre tâches, la tâche la plus prioritaire est celle ayant une valeur d'indice le plus faible.

3.1 Ordonnancement partitionné (2 points)

Est-il possible d'ordonnancer ce jeu de tâche avec une approche partitionnée en utilisant l'algorithme Rate Monotonic (RM) ? Pour répondre, représenter l'ordonnancement et indiquer l'instant éventuel de violation d'échéance par une des tâches.

3.2 Ordonnancement global (2 points)

Est-il possible d'ordonnancer ce jeu de tâches avec l'algorithme global-RM ? Pour répondre,

représenter l'ordonnement et indiquer l'instant éventuel de violation d'échéance par une des tâches.

3.3 Comparaison d'ordonnement [cours] (1 point)

Que nous enseigne ce contre-exemple en ce qui concerne l'approche globale et l'approche partitionnée en priorité fixe ?

4 Temps de blocage (5 points)

4.1 Protocole PIP [cours] (1.5 points)

Rappelez le fonctionnement du protocole PIP et appuyez vous sur un exemple. Donner le nombre de blocage en fonction du nombre de tâches moins prioritaires et du nombre de ressources utilisées.

4.2 Protocole PCP [cours] (1.5 points)

Rappelez le fonctionnement du protocole PCP et appuyez vous sur un exemple. Donner le nombre de blocage en fonction du nombre de tâches moins prioritaires et du nombre de ressources utilisées.

4.3 Mise en application des protocoles PIP et PCP (2 points)

On considère un système composé de 3 tâches T1, T2 et T3 qui partagent 2 ressources S1 et S2. T1 est plus prioritaire que T2 qui est elle-même plus prioritaire que T3. T1 (resp T2, T3) verrouille S1 pendant 3 unités de temps (resp. 2, 7) puis verrouille S2 pendant 4 unités de temps (resp 1, 5). Calculez les pires temps de blocage de T1 et T2 par des tâches de plus faibles priorités lorsque les protocoles de synchronisation sont PIP et PCP. Une attention particulière sera accordée aux explications, ce qui veut dire qu'un résultat correct non justifié ne donnera aucun point.

5 Couverture de code [cours] (3 points)

On considère le code suivant. Indiquer ce que doit faire un banc de tests qui cherche à faire de la couverture de code selon Statement Coverage, Decision Coverage et Modified Condition/Decision Coverage. Une attention particulière sera accordée aux explications.

```
if ((A && B) || C) {Ftrue}; Falways
```