Module STR
Examen session 2, 30 juin 2015
Systèmes temps réel
Durée : 1h20
Tous documents autorisés

Singhoff Frank
singhoff@univ-brest.fr
C-203

NB : ce document contient 2 feuilles recto/verso.

Exercice n^o 1 : Questions de cours (3 points)

– Question n^o 1.1 : 1 point

– Pourquoi modélise-t-on les fonctions d'un systèmes temps réel critique/dur par des tâches périodiques ?

– Question n^o 1.2 : 1 point

– Dans les système temps réel, on utilise presque toujours des algorithmes d'ordonnancement préemptifs. Pourquoi ?

– Question n^o 1.3 : 1 point

– Expliquez ce qu'est une inversion de priorité.

Exercice n^o 2 : Ordonnancement temps réel (6 points)

Dans cette partie, on regarde l'ordonnancement d'un système d'air conditionné pour un hôtel.

Ce dispositif est un boitier connecté à plusieurs ventilateurs. Un ventilateur est installé dans chaque chambre de l'hôtel et le boitier permet de régler de facon centralisée la température de toutes les pièces. Le boitier central comporte un processeur sur lequel plusieurs tâches s'exÃ©cutent. On suppose que les tâches sont périodiques et que l'on utilise un ordonnancement à priorité fixe preemptif. Dans l'environnement utilisé, les niveaux de priorités varient de 0 à 255. 255 est le niveau le plus élevé.

Nom	Capacité (secondes)	Période (secondes)
Ventilateur1	2	10
Ventilateur2	2	10
Clavier	3	15
LCD	1	5

1. Le tableau ci-dessus présente les différents paramètres des tâches du système. On étudie le système pour 2 chambres/ventilateurs. Affectez une priorité à chaque tâche. Expliquez comment vous affectez ces priorités.
2. Dessinez l'ordonnancement des tâches sur la période d'étude.
3. Montrez comment, **sans dessiner l'ordonnancement des tâches**, nous pouvions vérifier le respects des contraintes temporelles des tâches de cette application.
4. Combien de ventilateurs peut-on ajouter à ce jeu de tâches tout en respectant les échéances de toutes les tâches.

Exercice n^o 3 : Langage Ada (5 points)

Soit le programme suivant :

```
with Ada.Text_IO;                use Ada.Text_IO;
with Ada.Numerics.Float_Random;  use Ada.Numerics.Float_Random;

procedure Exercise is

   Generator : Ada.Numerics.Float_Random.Generator;

   task type Do_Nothing (Letter : Character);
   task body Do_Nothing is
      Period : Duration;
   begin
      -- Tirage aleatoire d'une valeur entiere
      -- (duree d'attente)
      Period := Duration (Random (Generator)) / 10;
      if Letter = 'W' or Letter = 'X' then
         Period := 2 * Period;
      end if;
      for Count in 1 .. 10 loop
         Put (Letter);
         delay Period;
      end loop;
   end Do_Nothing;

   type A_Ptr is access Do_Nothing;

   procedure Do_Nothing_More is
      type B_Ptr is access Do_Nothing;
      X : A_Ptr;
      Y : B_Ptr;
   begin
      X := new Do_Nothing (Letter => 'X');
      declare
         Z : Do_Nothing (Letter => 'Z');
      begin
         for Count in 1 .. 10 loop
            Put ('-');
            delay 0.008;
         end loop;
      end;
      Y := new Do_Nothing (Letter => 'Y');
   end Do_Nothing_More;

   W : Do_Nothing (Letter  => 'W');

begin
   Put_Line ("Calling Do_Nothing_More");
   Do_Nothing_More;
   Put_Line ("Returned from Do_Nothing_More");
end Exercise;
```

Questions :

1. Combien a-t-on de tâches dans ce programme ?
2. Pour chaque tâche x du programme, indiquer quelle est la tâche mère qui créée x.
3. Pour chaque tâche, indiquer si elle est allouée statiquement ou dynamiquement.
4. Est-on certain que rien ne sera affiché à l'écran avant l'affichage de la phrase *"Calling Do_Nothing_More"* ? Justifier votre réponse.
5. Est-on certain que la dernière chose à être affichée à l'écran sera *"Returned from Do_Nothing_More"* ? Justifier votre réponse.
6. Est-on certain que 'Y' sera affiché avant 'Z'. Justifier votre réponse.
7. Est-on certain que 'Y' sera affiché avant 'X'. Justifier votre réponse.