Université P. & M. Curie Examen Partiel
Licence Mention Informatique – L3 5 Novembre 2012
Programmation Concurrente (LI330) Durée : 1h45

Les téléphones portables doivent être éteints et rangés dans vos sacs.
Le mémento Ada/C est le seul document autorisé pendant l'épreuve.

Le barème (sur 20) est donné à titre indicatif.
Il vous est conseillé de soigner votre copie et de rédiger des réponses claires (en particulier les programmes).
Toutes vos réponses doivent être dûment justifiées.
Lisez attentivement le sujet. Toute réponse hors sujet sera considérée comme fausse.

Exercice 1 – Questions de cours (6 points)

Considérons le listing suivant :

```
1  with Ada.Text_Io;
2  use Ada.Text_Io;
3
4  procedure Ex_Tache is
5
6     task Serveur is
7        entry Je_Sers (Who : in Natural);
8     end Serveur;
9
10    task type Client (Id : Natural);
11
12    task body Serveur is
13
14    begin
15       Put_Line ("le serveur se lance");
16       loop
17          select
18             accept Je_Sers (Who : in Natural ) do
19                Put_Line ("le serveur sert" & Natural'Image (Who));
20             end Je_Sers;
21          or
22             terminate;
23          end select;
24       end loop;
25       Put_Line ("le serveur se termine");
26    end Serveur;
27
28    task body Client is
29    begin
30       Put_Line("client" & Natural'Image(Id) & " se lance");
31       Serveur.Je_Sers (Id);
32       Put_Line("client" & Natural'Image(Id) & " se termine");
33    end Client;
34
35 begin
36    declare
37       A : Client (2);
38       B : Client (1);
39       C : Client (3);
40    begin
41       null;
42    end;
43    declare
44       D : array (1 .. 2) of Client (4);
45    begin
46       null;
47    end;
48 end Ex_Tache;
```

Question 1 – 3 points
Construisez le chronogramme correspondant à l'exécution de ce programme. Vous indiquerez, pour les événements remarquables, le numéro des lignes correspondantes dans le fichier source.

Question 2 – 1 point
Donnez la définition d'une section critique. Quel(s) mécanisme(s) permet(tent) de la délimiter ?

Question 3 – 1 point
Quelle est la différence entre une ressource critique et une ressource banalisée ?

Question 4 – 1 point
Que signifie l'acronyme DHT ?

Exercice 2 – Grille de cumul (14 points)

Nous souhaitons développer une « grille de tâches » dont l'organisation logique suit le schéma de la figure 1. Les tâches y sont représentées par des cercles. Chaque tâche est identifiée par ses coordonnées sur la grille (x, y) où x représente le numéro de la ligne et y celui de la colonne. La *tâche d'origine* est située en haut à gauche (coordonnées 1,1). Les flèches indiquent le sens de communication des tâches entre elles.

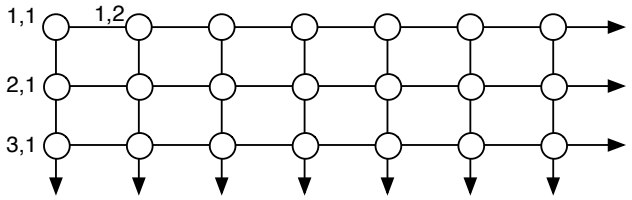

FIGURE 1 – Organisation des tâches du système

La tâche principale doit construire la structure et initier les communications en envoyant un message à la tâche d'origine. Ensuite, après avoir reçu ses coordonnées, chaque tâche doit propager tout message qu'elle reçoit à la fois vers la droite et vers le bas. Les tâches situées en dernière ligne (resp. dernière colonne) ne propagent pas le message vers le bas (resp. vers la droite). L'objectif est de comptabiliser le nombre de message propagés par chacune des tâches.

Lorsque les tâches de la grille ont terminé leur exécution, le programme principal demande à chacune d'afficher le nombre de message propagés. Pour notifier la fin de la propagation des messages au programme principal, la tâche en bas à droite de la grille accède à une variable partagée, encapsulée dans l'objet protégé `Attendre`. Cet objet protégé offre deux primitives d'accès :

- `Lancer_Affichage` : qui permet au programme principal de savoir quand il peut lancer l'affichage (*i.e.* les messages ont tous été émis et reçus),
- `Provoquer_Affichage` : qui permet à la dernière tâche [1] de la grille de notifier le programme principal qu'elle a reçu tous les messages qu'elle devait recevoir [2].

Question 1 – 1 point
Quel mécanisme (**`entry`**, **`procedure`** ou **`function`**) suggérez-vous pour implémenter les primitives d'accès `Lancer_Affichage` et `Provoquer_Affichage` ? Justifiez brièvement votre réponse.

Question 2 – 1 point
Pourquoi utilise-t-on un objet protégé pour cette communication ?

1. en bas à droite, elle a les coordonnées (`Coord_Max`, `Coord_Max`).
2. En considérant les paramètres déterminant le nombre de lignes et de colonnes dans le sujet, nous attendons, pour la tâche «en bas à droite», 924 messages.

Question 3 – 1 point
Proposez un type de données pour la variable protégée par l'objet `Attendre`.

Question 4 – 1 point
Ecrivez le source de la déclaration de l'objet protégé `Attendre`.

Question 5 – 1 point
Ecrivez le corps de l'objet protégé.

Nous disposons pour réaliser le programme des constantes et types suivants.

```
1   ------------------------------------------------------------------------
2   -- Les constantes
3   ------------------------------------------------------------------------
4   Coord_Max    : constant Positive := 7;
5   Valeur_Seuil : constant Positive := 924; -- Nombre de messages reçus par la tâche en bas à droite
6   ------------------------------------------------------------------------
7   -- Les types de base
8   ------------------------------------------------------------------------
9   type Coordonnees is range 1 .. Coord_Max;
```

Les tâches de la grille sont de type `Noeud`. Elles disposent de trois points d'entrées :
- un point d'entrée `Init` permettant au programme principal d'attribuer ses coordonnées à la tâche ; les paramètre sont les coordonnées (ligne, colonnes) de la tâche dans la grille,
- un point d'entrée `Contacter` permettant de recevoir un message (sans paramètre),
- un point d'entrée `Afficher` sans paramètre qui demande à la tâche d'afficher le nombre de messages qu'elle a reçu.

Question 6 – 1 point
Ecrivez le source de la déclaration du type de tache `Noeud`.

Nous souhaitons déclarer un tableau de `Noeud`s à deux dimensions, `Grille`. L'élément `Grille (i,j)` du tableau représente la tâche de coordonnées (i, j).

Question 7 – 0,5 point
Donnez le source déclarant la variable `Grille`.

Le programme principal : *1)* donne à chaque tâche ses coordonnées, puis, *2)* envoie un message à la première tâche, *3)* attends la notification d'affichage, et *4)* demande à chaque tâche d'afficher son compteur de messages.

Question 8 – 3 points
Ecrivez le source du programme principal.

Question 9 – 4,5 points
Ecrivez le corps d'une tâche de type `Noeud`.