SYSTÈMES ET RÉSEAUX INFORMATIQUES

COURS B4 : HTO(19339) et ICPJ(21937)
CYCLE PROBATOIRE INFORMATIQUE
(Conception et développement informatique)

EXAMEN DU 20 SEPTEMBRE 2001
partie portant sur l'enseignement des SYSTÈMES INFORMATIQUES

Date : jeudi 20 septembre 2001
Heure : 19h30 à 21h
Durée : 1 heure 30
Lieu : CNAM 35.2.25

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
Texte de 4 pages plus 3 pages d'annexes
Les annexes sont également fournies sous forme de feuille préremplie pour la solution

TOUS DOCUMENTS AUTORISES

<<<<<<<<<<
L'examen portant sur la partie systèmes informatiques comprend :
des questions sur la gestion des ressources
des questions sur la synchronisation

Chaque question peut-être traitée indépendamment.

Attention, vous n'avez pas de programme Ada à écrire (sauf la question 5), vous n'avez qu'à placer et initialiser des sémaphores et des variables globales.

Barème indicatif :
1. Ordre de démarrage : 2 points.
2. Allocation des ressources sans interblocage : 2 points.
3. Accès au canal partagé : 2 points.
4. Transfert des documents sur disque : 3 points
5. Communication par message : 2 points

Ne pas écrire au crayon ni à l'encre rouge
(sous peine de nullité, selon le règlement des examens)

>>>

Système avec des processus concurrents

On étudie quelques aspects du système suivant qui comprend 6 processus concurrents : le processus Main, 4 processus Acteurs, un processus Gestionnaire, et des paquetages de contrôle de concurrence. (c'est le système déjà étudié pour l'examen de juin. On en étudie d'autres aspects. Attention, on a changé quelques notations par endroits. Le corrigé de juin ne contient aucune solution qui puisse servir pour cet examen)

Les processus sont concurrents et les 6 processus démarrent en même temps. Chaque processus qui démarre, commence par demander l'impression d'une annonce. On suppose que chaque impression est une action atomique.

Chaque Acteur va exécuter un calcul compliqué, C, pour lequel il lui faut d'abord 2 ressources, puis 4 puis 7 ressources. A chaque cycle de calcul, l'Acteur produit un document qu'il envoie au gestionnaire G par le biais du paquetage Rapport. Le gestionnaire stocke sur disque les documents par paquets de 1 à 6 documents. Parallèlement avec ce stockage par le gestionnaire, chaque Acteur diffuse chaque document sur le réseau local. Pour cela il faut qu'il ait accès au canal de diffusion.

Le schéma du système est donné en fin du texte.

1. ORDRE POUR IMPRIMER LES ANNONCES DE DÉMARRAGE

Chaque processus qui démarre commence par demander l'impression d'une annonce. On suppose que chaque impression Put_Line est une action atomique (l'exclusion mutuelle est gérée par le sous-système d'entrée-sortie). Les processus sont concurrents et les 6 processus démarrent en même temps.

Le processus Main va orchestrer les impressions d'annonces de la façon suivante.

Chaque processus, sauf Main, commence par attendre le signal qui l'autorise à demander son annonce. Puis, quand il a reçu ce signal (qui est envoyé par Main), il imprime son annonce, et il envoie à son tour un signal vers Main.

De son coté, Main va les réveiller tous ensemble en envoyant à tous le signal qu'ils attendent. Puis Main attendra que tous les 5 autres processus lui signalent leur fin d'impression.

Question 1. Le paquetage Annonce fournit 4 procédures pour cette coordination :
- La procédure Attendre permet de bloquer le processus qui appelle cette procédure.
- La procédure LancerTous permet de libérer les 5 processus bloqués par Attendre ou en passe de l'être.
- La procédure Signaler permet d'envoyer un signal vers Main.
- La procédure AttendreLaFinDeTous permet à Main d'attendre les 5 signaux des autres processus.

Compléter la programmation du paquetage Annonce, donné en Annexe 1.

2. ALLOCATION DES RESSOURCES SANS INTERBLOCAGE

Chaque acteur demande successivement 1 ressource, puis 1 autre, puis 2 autres, puis 3 autres. Il obtient ainsi jusqu'à 7 ressources et il ne les restitue qu'à la fin. Quand il demande X ressources, et qu'elles ne sont pas disponibles, il n'en reçoit aucune et est bloqué.

Question 2.1. On veut ajouter un cinquième acteur, avec la même suite de demandes : 1, + 1, + 2, + 3. Combien faut-il de ressources au moins pour qu'il n'y ait jamais interblocage ?

Question 2.2. On veut garder les 4 acteurs, mais on veut changer leur code pour leur faire demander une ressource supplémentaire. La suite des demandes devient alors 1, + 1, + 2, + 3 + 1. Quel est le nombre minimum de ressources que doit avoir le système pour qu'il n'y ait jamais d'interblocage.

3. ACCÈS AU CANAL PARTAGÉ

Pour diffuser chaque document sur le réseau, chaque acteur doit avoir accès au canal. L'accès au canal partagé doit se faire en exclusion mutuelle entre les acteurs. Si on utilise classiquement un sémaphore d'exclusion mutuelle, on n'est pas maître de l'ordre des réveils en cas d'attente, car l'ordre de réveil dépend de l'ordre de la file d'attente du sémaphore, et cet ordre peut varier d'un système à un autre.

Pour pouvoir gérer explicitement l'ordre des réveils, on va utiliser le schéma des sémaphores privés et des variables d'état :
- une variable booléenne Libre pour indiquer que le canal est disponible,
- une variable booléenne Attente pour indiquer qu'il y a au moins un processus en attente
- un tableau de booléens Attend() pour indiquer quels sont les processus qui attendent.

La fonction ChoixCandidat élit un candidat selon la politique de réveil qui est programmée. On vous donne, dans l'annexe 2, une politique qui élit selon un ordre tournant.

Question 3.1. Compléter la programmation du paquetage Canal, donné en Annexe 2.

Question 3.2. On veut mettre en place un service selon la priorité des processus. On donne le plus petit nom au processus le plus prioritaire. Comment modifier le paquetage Canal pour que ce soit le processus de plus petit nom (plus petit Id) qui soit élu en premier quand il y a plusieurs processus en attente ?

4. TRANSFERT DES DOCUMENTS SUR DISQUE

Le gestionnaire lit un tableau qui contient Nb documents avec Nb compris entre 1 à 6. Chaque document doit être écrit sur un cylindre donné du disque. La valeur de ce cylindre est associée à chaque document et constitue la requête disque pour le document. Les Nb requêtes constituent un paquet de requêtes qui vont être traitées ensemble. Pour optimiser l'utilisation du bras porte-tête, le gestionnaire traite les paquets de requêtes d'écriture sur le disque selon le mélange suivant de politiques de déplacement des bras :
- si $1 \leq Nb \leq 3$, il sert les requêtes du paquet selon l'ordre de lecture des requêtes (ordre FIFO)
- si $4 \leq Nb \leq 6$, il réarrange les Nb requêtes du paquet selon la politique de l'ascenseur, en commençant selon le sens laissé par la dernière requête du paquet précédent, quelle que soit la taille du paquet et la politique appliquée pour ce paquet précédent.

Pour analyser cette façon de faire, on se donne une suite de 5 paquets de requêtes et on va comparer la politique mixte à ce que donnerait un service tout entier à l'ancienneté (FIFO).

On suppose que le bras est initialement sur le cylindre 10. Pour traiter le paquet suivant, on repart de la dernière position du bras. On a la suite :

Paquet	taille Nb	ordre de lecture des Nb requêtes	total des déplacements du bras quand tout le service est FIFO	réarrangement pour la politique mixte	total des déplacements du bras pour la politique mixte
S1	4	9, 8, 18, 3	1 + 1 + 10 + 15 = 27	9, 8, 3, 18	7 + 15 = 22
S2	5	7, 12, 2, 1, 4	4+5+10+1 +3 = 23	12, ….	
S3	6	12, 4, 6, 5, 7, 8			
S4	1	14			
S5	3	7, 9, 15			
Total			a =		b =

Question 4. Calculer (utiliser l'annexe 3) le total des déplacements du bras porte-têtes :
a) quand on se contente de servir les requêtes selon leur ordre d'arrivée
b) quand on applique la politique mixte
Quel gain obtient-on, en pourcentage, avec la politique mixte ? Que peut-on en dire sur le rapport coût/efficacité ?

5. CONTRÔLE DE CONCURRENCE PAR TÂCHE ADA SERVEUR

On a vu dans le cours qu'on pouvait utiliser une tâche serveur en Ada, avec des points d'appel ("entry") et avec invocation à distance (par message) par les clients, pour implémenter un paquetage de contrôle.

Question 5. Choisir un des paquetages ci-dessous et donner la programmation de son corps ("body") avec une tâche serveur.

```
--remplacer le paquetage Annonce par
task Annonce is
   entry Attendre;
   entry Signaler;
   entry LancerTous;
   entry AttendreLaFindeTous;
end Annonce ;

--remplacer le paquetage Canal par
task Canal is
   entry Ouvrir(I : in Integer);
   entry Fermer(I : in Integer);
   private
      entry Ouvrir2(I : in Integer);
      entry Ouvrir3(I : in Integer);
      entry Ouvrir4(I : in Integer);
      entry Ouvrir5(I : in Integer);
end Canal;
```

Système avec des processus concurrents

procedure Main is

type TabDoc is array(1..6) of Document;

```
-- contrôle l'ordre de mise en route - Q1
package Annonce is
   procedure Attendre;
   procedure Signaler;
   procedure LancerTous;
   procedure AttendreLaFindeTous;
end Annonce;
```

```
-- contrôle l'utilisation du canal partagé - Q3
package Canal is
   procedure Ouvrir(I : in Integer);
   procedure Fermer(I : in Integer);
end Canal;
```

```
-- contrôle l'allocation de ressources
package Ressource is
   -- X ressources au processus de nom I
   procedure Allouer(
                  X : in Integer; I : in Integer);
   -- Y ressources rendues par I
   procedure Restituer(
                  Y : in Integer; I : in Integer);
end Ressource;
```

```
-- contrôle l'archivage des documents
package Rapport is
   procedure Deposer(X : in Document);
   procedure Retirer(Y : out TabDoc);
end Rapport;
```

```
-- déclaration, création du Gestionnaire
task G is
   Id : Integer := 1; -- nom unique
   LesRapports : TabDoc; -- 6 documents
begin

   -- attendre le signal pour imprimer
   Annonce.Attendre;
   -- afficher son démarrage et son nom
   Put_Line(
      "Demarrage de " & Integer'Image(Id));
   -- annoncer qu'il a fini d'imprimer
   Annonce.Signaler;

   loop
      Rapport.Retirer(LesRapports);
      Stocker(LesRapports) --  sur disque
   end loop;

end G;
```

```
-- déclaration du type Acteur

task type TypeActeur is

   -- nom unique du processus Acteur
   Id : Integer := NomUnique; -- de 2 à 5

   --document établi par l'exécution de C
   MonCompteRendu : Document;
   begin

   -- attendre le signal pour imprimer
   Annonce.Attendre;
   -- afficher son démarrage et son nom
   Put_Line(
      "Demarrage de " & Integer'Image(Id));
   -- annoncer qu'il a fini d'imprimer
   Annonce.Signaler;

   Ressource.Allouer(1, Id);
   for K in 1..3 loop

      -- prendre K ressources de plus
      Ressource.Allouer(K, Id);
      C; --execution du code specifique
      --avec 2, puis 4 puis 7 ressources
      Rapport.Deposer(MonCompteRendu);

      Canal.Ouvrir(Id);
      Diffuser(MonCompteRendu); -- réseau
      Canal.Fermer(Id);

   end loop;

   -- rendre toutes les ressources
   Ressource.Restituer(7, Id);

end TypeActeur;
```

```
-- déclaration, création de 4 acteurs
A : array(2..5) of TypeActeur;
```

```
-- code du processus Main

   Id : Integer := 0; -- nom donné à Main
begin -- à ce point, les 6 tâches démarrent
   -- afficher son démarrage et son nom
   Put_Line("Demarrage de Main" );
   --autoriser les autres annonces
   Annonce.LancerTous;
   -- attendre la fin des annonces
   Annonce.AttendreLaFinDeTous;
   Put_line("Fin du demarrage");

end Main;
```

20 septembre 2001

ANNEXE 1
QUESTION 1 : ORDRE POUR LES ANNONCES DE DÉMARRAGE

```
package body Annonce is
   -- déclaration des sémaphores
   MainAAutres,AutresAMain : Semaphore;


   procedure Attendre is
   begin
      -- bloquer le processus appelant tant que Main n'a pas exécuté LancerTous


   end Attendre;

   procedure Signaler is
   begin
      -- informer le processus Main que l'appelant a fini


   end Signaler;

   procedure LancerTous is
   begin
      -- envoyer aux 5 processus le signal qui leur permet d'imprimer leur annonce




   end LancerTous ;
   procedure AttendreLaFinDeTous is
   begin
-- bloquer le processus Main tant que tous les 5 autres processus n'ont pas fait signe de vie




   end AttendreLaFinDeTous ;
begin
   -- initialiser chaque sémaphore



end Annonce;
```

ANNEXE 2
QUESTION 3 : ACCÈS AU CANAL PARTAGÉ

```
package body Canal is

   Libre : Boolean := True ; -- le canal est disponible quand Libre = True
   Attend : array(2..5) of Boolean:= False;  -- le processus I est bloqué quand Attend(I) = True
   Attente : Boolean := False; -- quand Attente = True, il y a au moins un processus en attente
   Suivant : Integer := 5;  -- nom d'un processus, index de parcours du tableau Attend
   -- déclaration des sémaphores utilisés


   procedure Ouvrir(I : in Integer) is
      -- le processus de nom I demande l'accès exclusif au canal
      OK : Boolean; -- accès possible ou non
   begin

      OK := Libre;
      if OK then
         Libre := False;  -- I prend le droit d'utiliser le canal
         -- décider qu'il faut autoriser I à continuer son exécution

      else
         Attente := True; Attend(I) := True;
         -- décider qu'il faut bloquer I en attente de ressource
      end if;

      -- appliquer la décision
   end Ouvrir;

   procedure Fermer(I : in Integer) is
      J : Integer;  -- nom du processus à réveiller
      function ChoixCandidat return Integer is -- choisit un processus en attente
      begin
         loop
            if Suivant = 5 then Suivant := 2 else Suivant := Suivant + 1; end if;
            exit when Attend(Suivant); -- sortie de la boucle  quand Attend(Suivant) = True
         end loop;
         return Suivant;
      end ChoixCandidat ;
   begin

      Libre := True; -- libération du canal par le processus I
      if Attente then
         J := ChoixCandidat; Attend(J) := False; Libre := False;
         -- réveiller le processus J

         -- voir s'il y a au moins un autre processus en attente et le noter dans Attente
         for K in 2..5 loop Attente := Attend(K); exit when Attente; end loop;
      end if;

   end Fermer;

begin          -- initialisation des sémaphores utilisés


end Canal ;
```

Examen de systèmes informatiques **numéro de copie :**
16 juin 2001

ANNEXE 3

QUESTION 4 : TRANSFERT DES DOCUMENTS SUR DISQUE

Le bras porte-têtes est initialement sur le cylindre 10.

Paquet	taille Nb	ordre de lecture des Nb requêtes	total des déplacements du bras quand tout le service est FIFO	réarrangement pour la politique mixte	total des déplacements du bras pour la politique mixte
S1	4	9, 8, 18, 3	1 + 1 + 10 + 15 = 27	9, 8, 3, 18	7 + 15 = 22
S2	5	7, 12, 2, 1, 4	4+5+10+1+3 = 23	12,	
S3	6	12, 4, 6, 5, 7, 8			
S4	1	14			
S5	3	7, 9, 15			
Total			a =		b =