SYSTÈMES ET RÉSEAUX INFORMATIQUES

COURS B4 : HTO(19339) et ICPJ(21937)

CYCLE PROBATOIRE INFORMATIQUE

(Conception et développement informatique)

EXAMEN DU 14 SEPTEMBRE 2004

portant sur l'enseignement des SYSTÈMES INFORMATIQUES

Date : mardi 14 septembre 2004
Heure : 19h45 à 21h15

Lieu : CNAM salles 35.2.25 et 35.3.26

TOUS DOCUMENTS PAPIERS AUTORISES

(les ordinateurs portables ne sont pas autorisés)

Texte d'énoncé de 4 pages
L'examen portant sur les systèmes informatiques comprend :
des questions sur la synchronisation des processus
des questions sur la gestion des ressources

Chaque question peut-être traitée indépendamment.

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

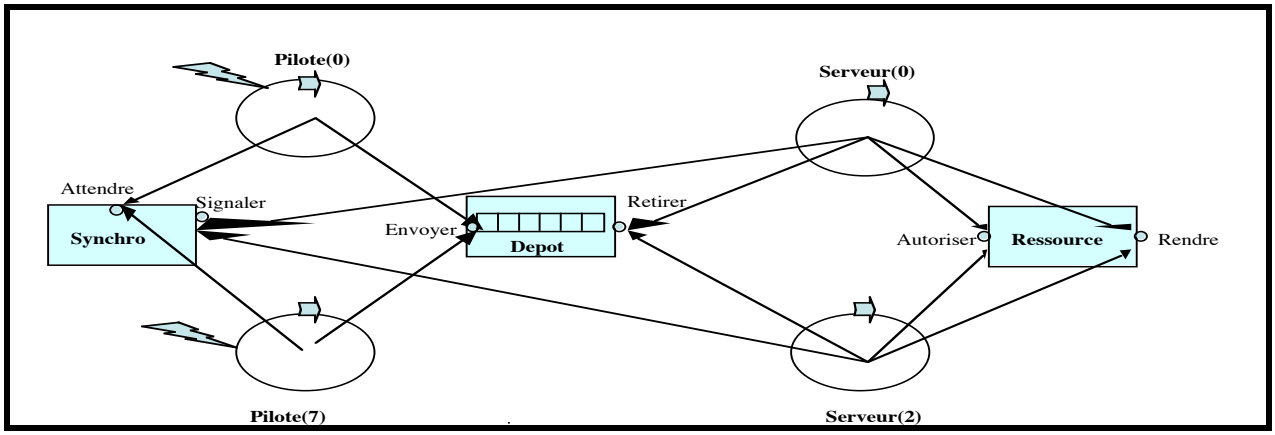
Barème *indicatif* pour la partie systèmes:

EXERCICE SUR LA SYNCHRONISATION DES PROCESSUS : (4 + 4 = 8 points)
EXERCICE SUR LES RESSOURCES : (4 points)

L'examen de septembre pour le cours B4 comprend deux parties de 1h30 chacune, une en réseaux et une en systèmes. Chaque partie compte pour 50% de la note finale

SYNCHRONISATION DES PROCESSUS

Un site Web utilise 8 appareils webcam de surveillance et récupère des images avec un lot de serveurs connectés à des clients. On a l'architecture suivante :




```
type Id_Cam is mod 8 ;

type Id_S is mod 3 ;

type Image is
  record
  Cam : Id_Cam ;
  Date : Time ;
  Paysage : Bitmap ;
end record ;

-- les processus coopérants

task type Un_Pilote ;
    Pilote : array(Id_Cam) of Un_Pilote ;

task type Un_Serveur ;
   Serveur : array(Id_S) of Un_Serveur ;
```

```
-- les paquetages de coopération

package Synchro is
  procedure Attendre(X : in Id_Cam);
  procedure Signaler(X : in Id_Cam);
end Synchro;

package Ressource is
  procedure Demander(X : in Id_Cam);
  procedure Rendre(X : in Id_Cam);
end Ressource;

package Depot is
  procedure Envoyer(X : in Id_Cam ; Y : in Image);
  procedure Retirer(X : in Id_Cam ; Y : out Image);
end Depot;
```

Chacun des trois processus Serveur est cyclique et est attaché à un client externe. Chaque Serveur commence par demander à son client le nom des caméras qu'il veut consulter soit A, B et C, (Ici on programme seulement le cas où la demande porte sur trois caméras). Après ce dialogue, le Serveur réserve chaque caméra demandée (c'est à dire A, B et C), puis il déclenche les pilotes associés et enfin retire du dépôt les images envoyées par ces pilotes pour les présenter au client.

Chacun des huit processus Pilote est cyclique et est attaché à une webcam. Chaque Pilote commence par attendre le signal de déclenchement, puis il lit la caméra et dépose l'image dans le tampon du dépôt. Clock est une fonction Ada qui retourne la date (et l'heure) courante.

```
task body Un_Pilote is
   X : Id_Cam := Nom.UniqueCam; Y : Image; Z : Bitmap ;
begin
   loop
        Synchro.Attendre(X) ; -- synchronisation pour attendre le signal envoyé à X par un serveur
        Lecture_de_la_Camera (Z) ;   -- acquisition de l'image Bitmap par un accès réseau local
        Y.Cam := X ; Y.Date : = Clock ; Y.Paysage := Z ;      -- préparation de Y
        Depot.Envoyer(X, Y) ; -- envoi vers les Serveurs de l'image Y prise par X
   end loop ;
end Un_Pilote ;
```

```
task body Un_serveur is
   I : Id_S := Nom.UniqueServeur ; A, B, C : Id_Cam; Y, Z, T : Image ; -- consultation de 3 webcam
begin
   loop
        Negociation_Avec_Le_Client(A, B, C) ; -- le client donne 3 noms de webcam
        Ressource.Demander(A) ; Ressource.Demander(B) ; Ressource.Demander(C) ;
        Synchro.Signaler(A) ; Synchro.Signaler(B) ; Synchro.Signaler(C) ;          -- réveil des pilotes
        Depot.Retirer(A, Y) ;  --  retrait de l'image de A qui est dans le dépôt et copie dans Y
        Depot.Retirer(B, Z) ;  --  retrait de l'image de B qui est dans le dépôt et copie dans Z
        Depot.Retirer(C, T) ;  --  retrait de l'image de C qui est dans le dépôt et copie dans T
        Envoi_Des_Images_Au_Client(Y, Z, T) ;
        Ressource.Rendre(A) ; Ressource.Rendre(B) ; Ressource.Rendre(C) ;
   end loop ;
end Un_Serveur ;
```

Question S1. Synchro et Ressource

Programmer les paquetages Synchro et Ressource en utilisant chaque fois un tableau de sémaphores. Ne pas oublier d'initialiser ces sémaphores, sinon le programme est faux et sera donc noté comme tel.
a) *Pour Synchro, utiliser Signal : array(Id_Cam) of Semaphore.*
b) *Pour Ressource, utiliser Mutex : array(Id_Cam) of Semaphore.*
c) *Montrer qu'il peut y avoir interblocage entre les serveurs. Pour cela considérer les trois demandes de caméras suivantes :*

a. Le Serveur(0) demande les caméras 2, 4, 1
b. Le Serveur(1) demande les caméras 4, 5, 6
c. Le Serveur(2) demande les caméras 6, 1, 2

d) *Donner deux méthodes qui permettraient d'éviter cet interblocage*

S2. Paquetage Depot.

On utilise les données persistantes

```
type Id_Depot is mod 6 ;
Tete, Queue : mod 6 := 0 ;
T : array (Id_Depot) of Image ;
```

La procédure Depot.Envoyer(X : in Id_Cam ; Y : in Image) permet aux pilotes de déposer des images à l'ancienneté dans le tampon du Depot. Le tampon peut contenir jusqu'à 6 images.

La procédure Depot.Retirer(X : in Id_Cam ; Y : out Image) doit permettre de retirer du tampon de Depot une image Y déposée par X, c'est à dire une image Y dont le champ Y.Cam est égal à X. Ce n'est pas un retrait à l'ancienneté. On veut cependant conserver l'utilisation du tampon sous forme de tampon circulaire habituel. Quand on est sûr que l'image Y (dont le champ Y.Cam est égal à X) est présente dans le tampon T, on l'extrait du tampon avec la procedure Extraire(X : in Id_Cam ; Y : out Image), qui est donnée ci-après.

Pour être sûr qu'une image de X est bien présente dans le tampon quand Extraire est appelée, on complète les procédures Depot.Envoyer(X : in Id_Cam ; Y : in Image) et Depot.Retirer(X : in Id_Cam ; Y : out Image). La procédure Depot.Envoyer(X : in Id_Cam ; Y : in Image) signale classiquement à l'ensemble des consommateurs l'arrivée d'un nouveau message, mais elle notifie aussi que l'envoi est fait par X. Cette notification est alors attendue dans Depot.Retirer(X : in Id_Cam ; Y : out Image) avant l'attente de la présence d'une image et la lecture par Extraire. On utilise un tableau de sémaphores pour contrôler cet envoi et cette attente particulière. On utilise un sémaphore de ce tableau dans Envoyer(X : in Id_Cam ; Y : in Image) et dans Retirer(X : in Id_Cam ; Y : out Image) pour permettre cette notification pour X. Soit pour ce tableau

```
Notification: array(Id_Cam) of Semaphore ;
```

On note qu'on a alors toujours la relation

nombre d'images dans le tampon =
nombre d'images issues du pilote 0 +
nombre d'images issues du pilote 1 +
… +
nombre d'images issues du pilote 7

```
procedure Extraire(X : in Id_Cam ; Y : out Image) is

-- on doit être certain qu'il y a dans le tampon T une image déposée par X
-- on va parcourir le tampon T à la recherche de T( ).Cam = X puis compacter le tampon si nécessaire
-- au retour de cette procedure, il y a une case libre de plus et elle est située à (Queue – 1) mod 6

        I : Integer ;
begin
        I := Tete ;                                     -- on commence la recherche en Tete
        while I /= Queue loop
                if T(I).Cam = X then
                        Y := T(I) ;
                        if (I + 1) mod 6 = Queue then return; end if ; -- on a retiré la dernière image
                        -- on a retiré une image et cela laisse un trou ; on remplit ce trou
                        while (I + 1) mod 6 /= Queue loop
                                T(I)  := T((I + 1) mod 6) ;      -- on déplace l'image suivante
                                I := I + 1 ;
                        end loop ;
                        return;  -- on a déplacé toutes les images qui étaient placées après Y
                end if ;
                I := (I + 1) modulo 6 ;          -- on n'a pas trouvé, on essaie la suivante
        end loop ;

-- Après ce retrait et cette gestion, il y a maintenant une case vide dans le tampon avant Queue
-- Depot.retirer doit donc mettre à jour Queue par l'instruction : Queue := (Queue – 1) mod 6 ;

end Extraire;
```

Question S2. Depot.Envoyer et Depot.Retirer

a) *Programmer le paquetage Depot en gérant la concurrence avec des sémaphores et en utilisant en particulier le tableau de sémaphores Notification et la procédure Extraire. Ne pas oublier d'initialiser ces sémaphores, sinon le programme est faux et sera donc noté comme tel.*
b) *Est-il vraiment nécessaire de signaler une nouvelle image à l'ensemble des consommateurs (et d'attendre cette notification globale)? Expliquer. Que déduire de cette explication?*
c) *Quel est le rôle et la valeur de la variable Tete? Qu'en déduire?*

REMPLACEMENT DE PAGE

R1 On dispose d'un système de mémoire paginée à la demande et de deux algorithmes A et B. On observe l'exécution d'un programme auquel le système alloue 3 cases de mémoire centrale et qui accède successivement aux pages : 2 3 5 2 1 5 2 4 5 3 2 5 2 3.

Chaque colonne représente, à un instant donné, de haut en bas, l'ordre de remplacement des pages en mémoire selon l'algorithme utilisé.

Avec l'algorithme A, on constate qu'il y a successivement en mémoire les pages suivantes :

état	2	2	2	3	5	2	1	5	2	4	5	3	3	5
après		3	3	5	2	1	5	2	4	5	3	2	5	2
chargement			5	2	1	5	2	4	5	3	2	5	2	3

Avec l'algorithme B, on constate qu'il y a successivement en mémoire les pages suivantes :

état	2	2	2	2	3	3	5	1	2	4	5	5	5	5
après		3	3	3	5	5	1	2	4	5	3	3	3	3
chargement			5	5	1	1	2	4	5	3	2	2	2	2

Question R1

a) *Lequel des deux algorithmes correspond à l'algorithme FIFO et lequel correspond à l'algorithme LRU? Justifier votre raisonnement*
Déterminer dans chaque cas le nombre total de défauts de page
b) *Quelles auraient été les pages en mémoire avec l'algorithme Optimal?*
et quel serait le nombre total de défauts de page?

R2 Un picosystème embarqué possède 4 cases de mémoire. Pour chacune des pages qui y sont placées, le système connaît la date de chargement, la date du dernier accès et l'état des indicateurs R de référence et M de modification (les dates sont données en top d'horloge) :

Page	Chargement	Dernier accès	R	M
0	120	280	1	0
1	200	265	0	1
2	100	270	1	0
3	110	285	0	1

Question R2

a) Quelle est la page qui sera remplacée avec l'algorithme LRU ?
b) Quelle est la page qui sera remplacée avec l'algorithme FIFO ?
c) Quelle est la page qui sera remplacée avec l'algorithme de la seconde chance ?