

# Systemes concurrents

2SN, janvier 2022

1h30, documents autorisés : notes et supports distribués en cours, TD, TP.  
Tous appareils électroniques prohibés, sauf autorisation expresse.

Les parties sont indépendantes. Barème sous réserve.

*Conseils :*

- il est généralement préférable de lire attentivement l'énoncé avant de répondre.
- résoudre un problème par sémaphores est souvent sensiblement plus chronophage et hasardeux que le résoudre par moniteurs ou rendez-vous...

## 1 Exécution concurrente (4 points)

Soit la proposition de protocole d'exclusion mutuelle suivante, où  $i = 0, 1$  désigne le numéro du processus  $P_i$  et  $j$  le numéro de l'autre processus :

*demande* : global array 0..1 of boolean := {false, false};  
*tour* : global 0..1 := 0;

(1) <i>demande</i> [ $i$ ] $\leftarrow$ true; (2) tant que <i>tour</i> $\neq i$ faire (3)     tant que <i>demande</i> [ $j$ ] faire nop; fintq (4) <i>tour</i> $\leftarrow i$ (5) fintq
---

(6)             section critique

(7) <i>demande</i> [ $i$ ] $\leftarrow$ false;
--

En supposant que les lectures et les écritures sont atomiques, montrer que ce protocole

1. n'est pas sûr, c'est-à-dire que  $P_0$  et  $P_1$  peuvent se trouver simultanément en section critique;
2. n'est pas vivace, c'est-à-dire qu'il est possible pour un processus de rester indéfiniment en attente pour entrer en section critique.

Pour cela, pour chaque question, dérouler clairement l'entrelacement des actions des processus  $P_0$  et  $P_1$  avec les valeurs des variables à chaque étape.

## 2 Vedettes, navettes, grippette (14 points)

Une compagnie maritime assure un ensemble de liaisons par bateau à partir d'un port d'attache. Elle dispose de 2 types de bateaux : des navettes ( $N$ ) et des vedettes ( $V$ ). L'équipage d'un bateau est constitué de 2 catégories d'employés : des marins ( $M$ ) et des personnels de bord ( $B$ ). Les effectifs de chaque catégorie dépendent du type de bateau : l'équipage d'une vedette doit exactement être constitué de  $Nb[M, V]$  marins et de  $Nb[B, V]$  personnels de bord (avec  $Nb[M, V]$  et  $Nb[B, V]$  strictement positifs). De même, l'équipage d'une navette doit exactement être constitué de  $Nb[M, N]$  marins et de  $Nb[B, N]$  personnels de bord (avec  $Nb[M, N]$  et  $Nb[B, N]$  strictement positifs).

Pour faire face à un manque de visibilité sur la disponibilité des personnels, suite à des mesures de quarantaine liées à une épidémie, une politique de gestion des équipages spécifique a été mise en place.

- Les personnels valides s'enregistrent selon leur catégorie lorsqu'ils arrivent au port.
- Lorsqu'un bateau est paré au départ, il appelle un équipage parmi les personnels disponibles. Attention :
  - L'appel de l'équipage est bloquant (et ne peut commencer) tant qu'un équipage complet n'est pas disponible.
  - La composition de l'équipage doit toujours respecter les contraintes d'effectif données par la matrice  $Nb[-, -]$
- Lorsqu'un personnel est effectivement appelé, c'est-à dire lorsqu'un bateau paré à partir peut disposer d'un équipage complet, il embarque sur le bateau.
- Lorsque le dernier personnel appelé est à bord, le bateau lève l'ancre.

### Remarques

- Le respect d'horaires ou la présence de passagers ne sont pas pris en compte et n'ont pas à être pris en compte pour cette politique.
- La compagnie dispose de suffisamment de places pour accueillir au port l'ensemble de sa flotte : il n'y a pas de conflit à gérer quant à l'accès au port ou aux places.

Les bateaux sont représentés par des processus ayant le comportement suivant :

Processus Bateau( $t$ ) //  $t$ :  $N$  ou  $V$  (Navette ou Vedette)

répéter

*appeler\_équipage( $t$ )* ;  
*lever\_ancre()* ;  
*naviguer()* ;  
*maintenance()*

sans fin

Les personnels sont représentés par des processus ayant le comportement suivant :

Processus Personnel( $c$ ) //  $c$ :  $M$  ou  $B$  (Marin ou personnel de Bord)

tant que *valide()* faire

*se\_présenter( $c$ )* ;  
*embarquer()* ;  
*travailler()* ;  
*débarquer()* ;

fin tant que

## Questions

3. (1 point) La mise en œuvre du protocole présenté est complétée par une contrainte supplémentaire, à savoir que **l'embarquement des équipages doit être réalisé en séquence** : si un embarquement est en cours, l'appel suivant ne peut être lancé que lorsque le bateau sur lequel l'embarquement est en cours a levé l'ancre. Expliquez précisément la raison (l'intérêt) de l'ajout de cette contrainte.

Donner le code des opérations `appeler_équipage(-)`, `lever_ancre()`, `se_présenter(-)`, `embarquer()` de manière à assurer les contraintes précédentes (et à garantir que dès lors qu'une ou plusieurs opérations sont possibles au regard des contraintes, au moins l'une de ces opérations est exécutée) dans les trois cas suivants :

4. (4 points) En supposant que l'on dispose de sémaphores (généraux), définis par une classe `Sémaphore`, fournissant **uniquement** les méthodes `up()`, `down()`, et un constructeur `Sémaphore(int vlnit)` permettant de fixer la valeur initiale du sémaphore à `vlnit`.
5. (4 points) En supposant que l'on dispose de moniteurs de Hoare (exécution des opérations automatiquement en exclusion mutuelle, priorité au signalé, files FIFO associées aux variables condition). On réalisera le moniteur en suivant la démarche vue en TD, basée sur l'évaluation de conditions d'état pour contrôler la progression des processus.
- (a) (1 point) Ecrire (en français) les conditions d'acceptation des opérations `appeler_équipage(-)`, `lever_ancre()`, `se_présenter(-)`, `embarquer()`
  - (b) (0,5 point). Définir les variables d'état permettant de représenter ces conditions.
  - (c) (0,5 point) Etablir un invariant du moniteur liant ces variables d'état
  - (d) (2 points) Programmer les opérations `appeler_équipage(-)`, `lever_ancre()`, `se_présenter(-)`, `embarquer()` du moniteur. Préciser (sous forme de commentaire dans le code) les pré/post conditions des signaler/attendre.
6. (3 points) En construisant une tâche serveur Ada, en suivant une méthodologie au choix (conditions d'acceptation ou automate).
7. (2 points) La mise en œuvre de ce protocole présente-t-elle un risque de famine dans le cas de l'utilisation
- des sémaphores généraux proposés
  - des moniteurs de Hoare

Justifiez précisément vos réponses, et dans le cas où il existe un risque de famine, indiquez un moyen de l'éviter.

## 3 Transactions (6 points)

8. Est-ce que le contrôle de concurrence continu par estampilles restreint davantage le parallélisme que le contrôle de concurrence par certification ? Justifiez votre réponse.



9. Le chronogramme suivant représente l'exécution de quatre transactions  $T_1, T_2, T_3, T_4$  opérant sur les variables partagées  $x, y$  et  $z$ . Les variables  $a, b, c, d$  sont locales à respectivement  $T_1, T_2, T_3, T_4$ . Dans ce chronogramme, le temps réel s'écoule de haut en bas, l'opération `tdébut()` lance une transaction, l'opération `tfin()` demande la validation de la transaction qui l'appelle.

- (a) Construire le graphe de dépendance correspondant à cette exécution.
- dans le cas où l'on utilise une stratégie de propagation directe des écritures ;
  - dans le cas où l'on utilise une stratégie de propagation différée des écritures ;
- Précisez pour chacun des arcs le numéro de l'opération entraînant la création de l'arc, et indiquer, en justifiant votre réponse, si cette exécution est sérialisable.

(0) initialement : ( $x := 0$ ) ( $y := 0$ ) ( $z := 0$ )				
	$T_1$	$T_2$	$T_3$	$T_4$
(1)	tdébut()			
(2)		tdébut()		
(3)			tdébut()	
(4)				tdébut()
(5)	$a := \text{tire}(z)$			
(6)		$b := \text{tire}(y)$		
(7)			técrire( $z, 1$ )	
(8)	técrire( $x, 2$ )			
(9)				técrire( $y, 3$ )
(10)			$c := \text{tire}(y)$	
(11)		técrire( $z, 4$ )		
(12)				$d := \text{tire}(x)$
(13)			tfin()	
(14)		tfin()		
(15)				tfin()
(16)	tfin()			

- (b) On suppose que l'exécution des transactions est contrôlée selon un protocole de contrôle de concurrence. L'ordre d'exécution local à chaque transaction reste le même (tant que la transaction n'est pas abandonnée) mais l'ordre relatif des opérations de transactions distinctes peut être différent, dans la mesure où lorsqu'une transaction est bloquée, ses opérations sont retardées.
- Indiquez quel sera le résultat final de cette exécution avec un protocole de contrôle de concurrence par certification, et une propagation différée des écritures : dernière valeur des variables  $x, y$  et  $z$ , transactions validées/abandonnées/bloquées. Précisez quel sera l'ordre série équivalent obtenu.
  - Même question dans le cas d'un contrôle de concurrence continu par estampilles, en supposant que les estampilles sont attribuées selon l'ordre de lancement des transactions.
  - Même question dans le cas d'un contrôle de concurrence continu par verrouillage à 2 phases strict.