Université Larbi ben M'hidi-Oum El Bouaghi
Faculté des Sciences Exactes et Sciences de la Nature et de la Vie
Département de Mathématiques et Informatique

OEB le 21Octobre 2017

Concours national d'accès à la formation de troisième cycle (doctorat LMD)

Epreuve 1 : Algorithmes Distribués (Sujet 2)

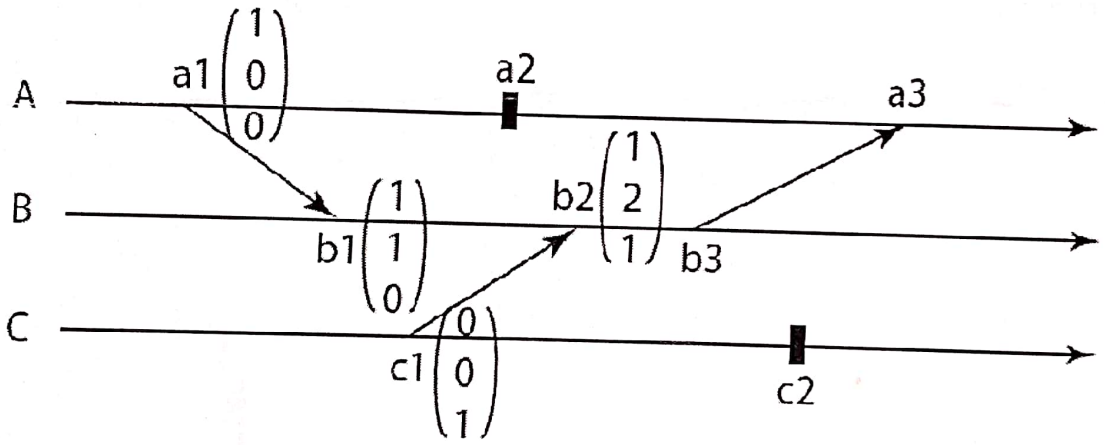
Durée : 2h

Exercice1 :Questions de compréhension (06pts)

1. Quel est l'intérêt de la synchronisation des horloges des différents sites dans un système réparti ?
2. Que signifient « un canal fifo » , « un canal fiable »?
3. A quoi sert le protocole de remise à zéro dans certains algorithmes répartis ?

Exercice2 :(6 pts)

Le chronogramme de la figure ci-dessous représente une partie d'une exécution répartie. Les événements sont datés à l'aide d'un système d'horloges vectorielles de Mattern.



Questions :

1. Donnez pour chacun des évènements sa date correspondante.
2. Que représente la somme des éléments d'un vecteur 'horloge' ?
3. En se basant sur les dates des événements, indiquez si les événements a2 et c2 sont causalement dépendants ou non. Justifiez vos réponses.
4. En se basant sur les relations causales, quel est le plus long chemin dans cette exécution?

Exercice 3 : (8 pts)

Dans un système réparti, on considère n sites (processus) et un ensemble de M ressources identiques, chacune devant être utilisée en section critique. Les demandes des sites pouvant concerner un nombre quelconque k de ressources ($1 \leq k \leq M$). Soit l'algorithme de Raynal d'allocation répartie de ces ressources suivant :

- Ri : ensemble des sites auxquels le site i doit demander l'autorisation.
- $utilisé_i[j]$: la perception qu'a le site i du nombre de ressources utilisées par j
- $différé_i$: ensemble des sites vers qui le site i retarde l'envoie de réponse.
- libre(y) : message indiquant que le site utilise M-y ressources.
- hi : valeur de l'horloge locale

Comportement d'un site i :

Lors d'un appel à acquérir :

ki := nombre de ressources demandées par le site i
$état_i$:= demandeur ;
$last_i := h_i + 1$;
$\forall j \in Ri$: **si** $utilisé_i[j] = 0$ **alors**
$utilisé_i[j] := M$;
$demandé_i[j]$:= vrai ;
envoyerrequête($last_i$, i) à j ;
sinon$demandé_i[j]$:= faux ;
fsi ;
si $(\sum_{1 \leq j \neq i \leq n} utilisé_i[j] + k_i) \leq M$ **alors**$état_i$:= dedans **fsi** ;
attendre ($état_i$ = dedans) ;

Lors de la réception de requête(k, j):

$h_i := \max(h_i, k)$;
$priorité_i$:= ($état_i \neq$ dehors) et ($last_i$, i) < (k, j) ;
si**non**$priorité_i$**alors envoyer** libre(M) à j ;
sinon
si $k_i \neq M$ **alorsenvoyer** libre(M - k_i) à j **fsi;**
$différé_i := différé_i \cup \{ j \}$;
fsi

Lors d'un appel à libérer :

$état_i$:= dehors;
$\forall j \in différé_i$: **envoyer**libre(k_i) à j ;
$différé_i := \phi$;

Lors de la réception de libre(y) venant de j

$utilisé_i[j] := utilisé_i[j] - y$;
si$état_i$ = demandeur **alors**
si($utilisé_i[j] = 0$ et **non**$demandé_i[j]$) **alors**
$utilisé_i[j] := M$;
envoyerrequête($last_i$, i) à j ;
$demandé_i[j]$:= vrai ;
sinon si $(\sum_{1 \leq j \neq i \leq n} utilisé_i[j] + k_i) \leq M$ **alors**
$état_i$:= dedans;
fsi
fsi
fsi

Questions :

1. L'algorithme permet-il d'éviter les interblocages ? Justifiez vos réponses.
2. Peut-on considérer que cet algorithme est un algorithme adaptatif ?
3. Que signifie : $demandé_i[j]$ = vrai et $utilisé_i[j] = 0$
4. Que signifie : $demandé_i[j]$ = faux et $utilisé_i[j] = 0$
5. Lors de la réception de requête(k, j), quel est l'intérêt du test :**si**ki≠ M
6. Etudiez la complexité de l'algorithme en termes de nombre de messages circulés

Bon courage!