

Examen semestriel

Algorithmique et Systèmes d'exploitation distribués  
1 H30

**Exercice 1 (10 points)** : Répondre aux questions suivantes.

**Question 1** : Qu'est ce qui motive le recours aux architectures parallèles ?

**Question 2** : Il y'a quelques années , il était impossible de faire des prévisions météorologiques de plus de 3 jours. Expliquez pourquoi.

**Question 3** : Le Web peut-il être considéré comme un système réparti ? Justifiez.

**Question 4** : Pour gérer l'allocation des ressources dans un système réparti, deux méthodes sont possibles :

- Méthode 1 : On désigne un seul site allocateur qui est chargé de gérer toutes les ressources ; qui sont dispersées sur les différents sites.
- Méthode 2 : On choisit plutôt de répartir la gestion des ressources entre plusieurs allocateurs qui peuvent être présents sur des sites différents.

Donnez les avantages et les inconvénients de chaque méthode.

**Question 5** : Le problème du consensus est un problème réputé difficile dans les systèmes répartis. Expliquez brièvement de quoi il s'agit.

**Exercice 2 (10 points)** : On considère l'algorithme d'élection de processus suivant.

Chaque processus possède une capacité représentée par un nombre  $C_i$ . Le processus à élire doit posséder la plus grande capacité. Chaque processus peut être dans l'un des états suivants :

- « Actif » : le processus est en activité.
- « Candidat » : le processus est candidat à l'élection.
- « Perdu » : le processus n'a aucune chance d'être élu.
- « Elu » : le processus est élu. Il doit être proclamé.

**Algorithme :**

Initialement, tous les Processus sont dans l'état « Actif ».

Le processus initiateur de l'élection  $P_i$  diffuse le message  $Elire(C_i, i)$  à tous les processus et passe à l'état « Candidat ».

A la réception d'un message  $Elire(C_k, k)$ , un processus  $P_i$  :

s'il est dans l'état « Actif » :

- il répond  $Ack(i)$  à  $P_k$  si sa capacité  $C_i$  est inférieure ou égale à  $C_k$ .
- il émet  $Elire(C_i, i)$  en diffusion si sa capacité  $C_i$  est strictement supérieure à  $C_k$  et passe dans l'état « Candidat »

s'il est dans l'état « **Candidat** » :

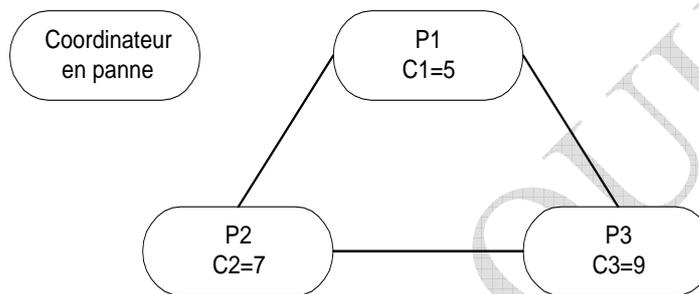
- il ignore ce message si sa capacité  $C_i$  est supérieure à  $C_k$ .
- il répond Ack(i) à  $P_k$  si sa capacité  $c_i$  est inférieure ou égale à  $C_k$  et passe dans l'état « Perdu » (ne peut pas être élu, il supprime alors tous les Ack qu'il a déjà reçu).

s'il est dans l'état « **Perdu** » :

- il répond Ack(i) à  $P_k$  si sa capacité  $C_i$  est inférieure ou égale à  $C_k$ .

**Question 1** : Avec cet algorithme, quand un processus peut-il savoir s'il est le nouvel élu ?

**Question 2** : La figure suivante représente un système où le processus P1 a détecté que le processus coordinateur est tombé en panne .



Complétez le tableau suivant montrant le déroulement de l'algorithme pour l'élection d'un nouveau processus coordinateur.

Processus	Etat	Message reçu	Nouvel état	Action
P1	Actif		Candidat	Diffuser Elire (5, 1)
...	.....	.....	.....	.....
...	.....	.....	.....	.....
...	.....	.....	.....	.....

**Question 3** : Calculer la complexité de cet algorithme, en nombre de messages. Justifiez.

**Question 4** : Proposez une méthode pour réaliser l'action « Proclamer ».

**Question 5** : Que doit faire un processus qui reprend son exécution après une panne ? Quelles conséquences peuvent survenir après son action ?