

Commencé le	jeudi 2 avril 2020, 15:02
État	Terminé
Terminé le	jeudi 2 avril 2020, 16:27
Temps mis	1 heure 24 min
Note	15,00 sur 20,00 (75%)

Description

Quiz concernant :

Votre note est disponible immédiatement à la fin du quiz, mais la correction est disponible uniquement après la fermeture du test.

Les questions à choix multiples disposent de réponses fausses à points négatifs.

Question 1

Terminer

Note de 2,00 sur 2,00

Le temps pour un ordinateur

Veuillez choisir au moins une réponse :

- ☒ a. Le circuit RTC est lu au moment du démarrage pour savoir l'heure.
- ☐ b. Avec un algorithme approprié, il n'y a aucune difficulté à synchroniser l'heure de deux ordinateurs avec autant de précision que désiré, en autant qu'ils soient connectés en réseau, quelles que soient les caractéristiques du réseau.
- ☐ c. Les ordinateurs viennent avec une horloge précise, le temps exact est configuré en usine, au moment de l'assemblage, et il n'y a aucun besoin de corriger l'heure par la suite.
- ☒ d. Une fois démarré, l'ordinateur configure une minuterie pour se faire interrompre à intervalle régulier afin de tenir un décompte du temps.

Question 2

Terminer

Note de 1,33 sur 2,00

Distribution de contenu

Veuillez choisir au moins une réponse :

- ☒ a. Le serveur peut propager les mises à jour à tous les réplicats, voire même à tous les clients. Ceci est utile si chaque récipiendaire est presque certain d'avoir besoin de toutes les mises à jour (par exemple les mises à jour de sécurité mensuelles pour les postes de travail).
- ☒ b. Le serveur peut envoyer un message d'invalidation à tout réplicat (incluant les clients avec cache) qui possède une copie qui était à jour de la variable qui est mise à jour. Ceci est utile lorsque les mises à jour sont très fréquentes et chaque client possède un petit sous-ensemble de valeurs à jour.
- ☐ c. Le serveur peut associer une date de validité à tout élément de donnée (e.g. entrée DNS ou page Web). Ceci est utile si les clients veulent grader ces informations en cache, les mises à jour ne sont pas trop fréquentes et chaque client utilise une très petite fraction des mises à jour.
- ☐ d. Le mieux est d'envoyer toutes les mises à jour à tous les réplicats. C'est la méthode la plus performante et la plus simple.

Question **3**

Terminer

Note de 2,00
sur 2,00

Cache dans les clients

Veuillez choisir au moins une réponse :

- ☒ a. Les clients du système de fichiers NFS utilisent une cache au niveau du client mais seulement pour quelques secondes, ce qui évite la plupart des conséquences perceptibles des problèmes de cohérence.
- ☐ b. Les clients ne peuvent jamais maintenir une cache, sauf s'ils sont enregistrés au niveau du serveur et ont l'assurance que le serveur va les notifier, quel que soit le modèle de cohérence qu'ils veulent offrir.
- ☒ c. Les fureteurs Web utilisent souvent une cache au niveau du client.
- ☐ d. Les caches au niveau du client, par exemple sur le Web, sont une fausse bonne idée car la plupart des pages Web ne sont lues qu'une seule fois alors qu'avec une cache il faut deux fois les accéder deux fois, une fois pour les afficher et une autre fois pour prendre une copie en cache.

Question **4**

Terminer

Note de 1,00
sur 2,00

Election hiérarchique

Veuillez choisir au moins une réponse :

- ☒ a. Les participants ont un ordre de priorité pré-établi.
- ☐ b. L'élection hiérarchique ne fonctionne que s'il n'y a aucun participant en panne.
- ☒ c. Lorsqu'un participant ne parvient pas à contacter celui qu'il considère l'élu, il déclenche une élection et demande dans l'ordre aux participants plus prioritaires.
- ☒ d. L'intérêt de l'élection hiérarchique est qu'on peut garantir qu'il n'y a toujours qu'un seul élu, quelles que soient les pannes de noeud ou de réseau, étant donné qu'il y a un ordre pré-établi.

Question **5**

Terminer

Note de 2,00
sur 2,00

L'exclusion mutuelle

Veuillez choisir au moins une réponse :

- ☒ a. L'exclusion mutuelle assure qu'un seul processus peut effectuer une certaine opération pendant l'intervalle où il possède le verrou associé.
- ☐ b. L'exclusion mutuelle en anneau est sécuritaire, vivace et respecte l'ordre premier arrivé, premier servi.
- ☐ c. L'exclusion mutuelle par serveur central est à éviter car elle est moins efficace et plus fragile que celle en anneau.
- ☒ d. Un système d'exclusion mutuelle est vivace si aucun client ne peut voir sa requête attendre indéfiniment, alors que les autres réussissent à passer avant lui.

Question **6**

Terminer

Note de 2,00
sur 2,00

Les méthodes de synchronisation

Veuillez choisir au moins une réponse :

- ☐ a. La méthode de Berkeley est un algorithme simple pour synchroniser l'heure d'un client sur l'heure d'un serveur.
- ☒ b. Une horloge GPS, qui produit un signal de synchronisation Pulse Per Second, est un très bon moyen pour synchroniser précisément l'heure d'un ordinateur, par exemple pour un serveur de temps primaire sur Internet.
- ☒ c. La méthode de synchronisation utilisée dans le protocole NTP est semblable à l'algorithme de Christian, à la différence qu'on soustrait le temps passé sur le serveur pour répondre. Ceci permet de ne conserver idéalement que le temps passé sur le réseau comme incertitude.
- ☒ d. La méthode de Christian est un algorithme simple pour synchroniser l'heure d'un client sur l'heure d'un serveur.

Question 7

Terminer

Note de 0,67
sur 2,00**La cohérence séquentielle**

Veuillez choisir au moins une réponse :

- ☒ a. Tous les processus voient éventuellement les mêmes valeurs écrites dans le même ordre dans la base de donnée centrale (mémoire partagée).
- ☐ b. Ce modèle de cohérence de données a été défini par Leslie Lamport, le même qui a proposé les horloges logiques et publié le système d'édition LaTeX.
- ☒ c. Plusieurs processus concurrents qui accèdent les mêmes variables verront exactement les mêmes valeurs en même temps pour chaque variable.
- ☒ d. Un même processus voit ses accès de lecture et écriture effectués dans l'ordre dans lesquels il les a effectués.

Question 8

Terminer

Note de 2,00
sur 2,00**Protocoles pour la réplication**

Veuillez choisir au moins une réponse :

- ☒ a. Un serveur primaire qui reçoit toutes les écritures permet de les sérialiser et de s'assurer que le même ordre d'écriture arrivera sur tous les réplicats, si le serveur primaire propage, dans l'ordre, les écritures reçues.
- ☐ b. Les clients peuvent sans problème écrire sur le serveur de leur choix, en autant que le serveur note l'heure de la modification. Les serveurs répliqués se propagent ensuite les mises à jour, en tenant compte de l'heure de modification. Ceci assurera quand même une cohérence séquentielle aux clients, autant pour les lectures que les écritures.
- ☒ c. Un système de votation pour l'acceptation des mises à jour est plus complexe mais assure que le système peut fonctionner même en cas de pannes.
- ☒ d. Chaque client peut écrire ses modifications sur l'ensemble des réplicats. Cependant, pour assurer un ordre cohérent, il faut qu'il y ait un verrou qui assure qu'il n'y a pas de mises à jour concurrente des mêmes données, ou qu'il y ait un serveur primaire qui détermine l'ordre des mises à jour concurrentes.

Question 9

Terminer

Note de 0,67
sur 2,00**- Lectures et écritures comme ensemble ordonné (monotonie)**

Veuillez choisir au moins une réponse :

- ☒ a. La monotonie des lectures veut dire qu'après avoir vu une nouvelle valeur pour une variable, un même processus ne pourrait plus voir une valeur plus ancienne par la suite.
- ☐ b. Si un client peut se connecter à différents serveurs répliqués, et que les mises à jour peuvent être retardées sur certains serveurs par rapport à d'autres, le client pourrait voir une nouvelle valeur pour une variable sur un serveur A et un peu plus tard voir une valeur plus ancienne pour la même variable sur un serveur B.
- ☒ c. La cohérence séquentielle implique une monotonie des lectures et des écritures.
- ☒ d. La monotonie des écritures implique nécessairement la monotonie des lectures.

Question 10

Terminer

Note de 1,33
sur 2,00**L'état global d'un système réparti**

Veuillez choisir au moins une réponse :

- ☒ a. L'état de chaque processus peut être vu comme ce qui résulte de son exécution depuis son initialisation, et peut être représenté comme l'ensemble des événements (messages) qu'il a reçus.
- ☐ b. Pour obtenir l'état global d'un système réparti, il suffit d'envoyer un message à tous de mémoriser le numéro d'événement de la réception de ce message.
- ☒ c. L'état global est l'union de l'état de chaque processus du système.
- ☐ d. Si on est prêt à arrêter tous les processus pendant un certain intervalle, il est assez facile de prendre l'état global du système pendant cet intervalle.