

Examen de Systèmes Répartis
15 décembre 2003

Durée :2 heures ; tous documents autorisés

N.B. L'examen comporte 4 problèmes indépendants. Prière de lire attentivement l'ensemble de l'énoncé avant de commencer à répondre, et de respecter les notations du texte. La longueur de l'énoncé n'est pas un signe de difficulté, mais est nécessaire pour bien spécifier les problèmes. La **clarté**, la **précision** et la **concision** des réponses, ainsi que leur **présentation matérielle**, seront des éléments importants d'appréciation.

Problème 1. (6 points).

Les questions sont indépendantes. On demande des réponses brèves mais précises.

- 1) Dans le protocole d'appel de procédure à distance, expliquer pourquoi il est difficile de transmettre des pointeurs en paramètre. Indiquer des moyens possibles pour surmonter cette difficulté.
- 2) Dans le protocole d'appel de procédure à distance, expliquer avec précision le rôle et l'utilisation du *portmapper*.
- 3) Indiquer l'intérêt, dans un appel de procédure à distance, d'avoir un appel idempotent. Donner un exemple d'appel idempotent et un exemple d'appel non idempotent réalisant la même opération.
- 4) Expliquer la notion de "fabrique" (*factory*) et son intérêt dans les systèmes répartis à objets. Y a-t-il des différences entre Corba et DCOM dans la mise en œuvre de cette notion ?
- 5) Dans le protocole Kerberos, pourquoi introduit-on des authentificateurs, alors que les tickets assurent déjà une protection ?
- 6) Soit deux correspondants A et B qui communiquent entre eux en utilisant un protocole de cryptographie à clé publique. On note respectivement KPA et KPB les clés publiques de A et B, KSA et KSB les clés privées correspondantes de A et B. On considère les 4 messages ci-après, envoyés par A à B (où M désigne un message « en clair » non spécifié) :
 - 1) $\{M\}_{KPB}$
 - 2) $\{M\}_{KSA}$
 - 3) $\{\{M\}_{KSA}\}_{KPB}$
 - 4) $\{\{M\}_{KPC}\}_{KPB}$ (C est un correspondant distinct de A et B et KPC est sa clé publique)

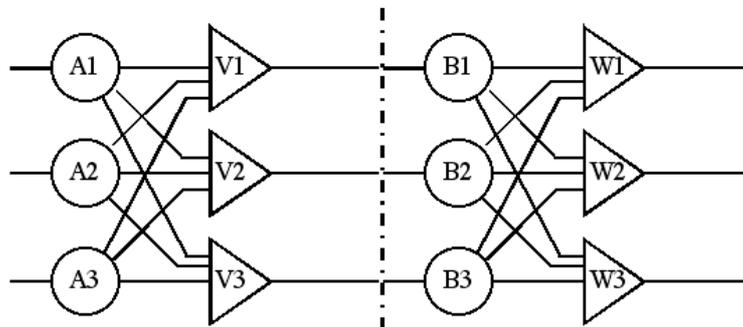
Pour chacun de ces messages, dites quelle est d'après vous sa fonction, quelles garanties éventuelles il procure, et expliquez comment il sera traité par B lors de sa réception.

Problème 2 (4 points)

La gestion des caches dans le système de fichiers répartis NFS ne réalise pas la cohérence de manière rigoureuse.

- 1) Expliquer ce mécanisme et indiquer précisément où se situe le problème.
- 2) Donner un scénario précis d'utilisation, dans lequel le déroulement d'une série d'accès à un fichier par deux processus donne des résultats différents sur une machine unique et sur deux machines différentes.
- 3) Indiquer pourquoi, dans les utilisations pratiques, on s'accommode de cette situation.

Problème 3 (4 points)



Un système tolérant les fautes est construit selon le principe de la redondance modulaire triple selon le schéma de la figure ci-dessus. Il y a 2 étages de traitement A et B (reproduits en 3 exemplaires) et 3 voteurs majoritaires par étage. La fonction du système est d'effectuer en séquence les opérations A et B sur une entrée unique dépendante du temps, pour fournir une sortie unique. On suppose que les défaillances sont des pannes franches.

Question 1. Définir précisément ce qu'on entend par "fonctionnement correct" pour ce système. Quel(s) élément(s) faut-il ajouter pour l'utiliser effectivement ?

Question 2. Indiquer pour chaque hypothèse de défaillance ci-après si le système peut maintenir un fonctionnement correct (justifier votre réponse) :

Panne de A1 et B1

Panne de A1 et A2

Panne de V1 et B2

Panne de V1 et W2

Question 3. Indiquer avec précision un scénario où le système fonctionne correctement avec le nombre maximal possible d'éléments défaillants. Quel est ce nombre ?

Question 4. Dans un système dont le degré de redondance est 5 au lieu de 3, quel nombre maximal d'éléments et de voteurs par étage peuvent-ils être simultanément défaillants tout en assurant un fonctionnement correct ?

Problème 4 (6 points)

Ce problème concerne une application dans laquelle un ensemble d'utilisateurs collaborent à un travail commun, la rédaction d'un document. Chaque personne dispose d'une station indépendante et ces machines communiquent par un réseau.

On dispose d'un système de communication par événements de type *publish-subscribe*. On peut définir des sujets (que l'on notera par des identificateurs), et les opérations sont les suivantes :

subscribe(subject) - s'abonner au sujet *subject*.

unsubscribe(subject) - se désabonner du sujet *subject*.

publish(event, subject) - émettre un événement *event* sur le sujet *subject*. On peut associer une information quelconque (par ex. fichier) à l'événement *event*.

attach(handler, subject) - déclaration d'un traitant *handler* (un programme) qui sera activé lors de la réception d'un événement sur le sujet *subject*.

On suppose par ailleurs disponible un système partagé de gestion de fichiers où chaque utilisateur peut définir des fichiers privés (accessibles à lui seul) et des fichiers partagés (accessibles à tous).

Question 1. Le document est divisé en parties, et chaque personne est responsable de l'édition d'une partie : elle est la seule à modifier cette partie. A certains points de la rédaction (validation), un rédacteur peut rendre l'état de sa partie visible aux autres. Ce nouvel état sera alors visible à tous sans modification, jusqu'à la prochaine validation.

On demande de décrire avec précision l'architecture d'ensemble de l'application, en indiquant comment sont gérées les données et comment se déroulent les opérations.

Question 2. On souhaite maintenant rendre ce système tolérant aux fautes, pour éviter de perdre des données et pour permettre en permanence l'accès au document. Indiquer avec précision le principe d'une solution. L'hypothèse de défaillance est la panne franche d'une machine.

Question 3. On considère maintenant une réalisation différente de l'application de gestion de documents. Chaque personne édite sa partie de manière privée. De temps en temps, elle diffuse ses modifications à tous ; chacun met alors à jour sa version du document. Il n'y a pas d'informations partagées.

On demande de décrire avec précision l'architecture d'ensemble de l'application. Quelle propriété doit avoir le mécanisme de diffusion d'événements pour préserver la cohérence du document ?

Question 4. Comparer les deux principes de réalisation proposés dans les questions 1 et 3 (avantages et inconvénients).