Examen Client/Serveur, 1996/1997

Tout document autorisé

Lire le sujet en entier avant de commencer.

Un traitement de texte

Ce qui nous intéresse ici est de définir une architecture ouverte, propre et évolutive, et non pas quelque chose de rapide et figé.

On considère que les principales fonctionnalités d'un traitement de texte sont :

- Le chargement et la sauvegarde de fichiers avec différents formats.
- L'impression.
- La visualisation *WYSIWYG* du texte sur l'écran.
- La correction orthographique.
- L'édition du texte.
- Le choix d'attributs d'affichage (gras, italique, fontes, ...).
- La possibilité de définir des fonctions plus élaborées (macros)
- La possibilité de définir des styles.
- L'inclusion d'images, tableaux, formules mathématiques, ...
- La traduction du texte dans une autre langue..
- Le *Undo/Redo*
- La possibilité de faire des *batchs* (lancer une impression sans lancer l'interface utilisateur)
- Barre d'outils : taille fonte, alignement, tabulation, style, ...
- Interaction avec l'utilisateur
- ...

1 A partir de cette liste de fonctionnalités dans le désordre, proposez une architecture client/serveur ou l'on éclate en de nombreux processus l'application traitement de texte.

Pour simplifier, essayez de diminuer le nombre de protocoles différents utilisés (sans néanmoins faire un protocole "fourre-tout")

Indiquez bien le type de chaque processus (client, serveur, client/serveur) ainsi que le protocole qu'il utilise. Sans oublier les flèches pour indiquer qui utilise qui.

Détaillez les protocoles que vous utilisez.

2 L'implémentation des protocoles, est-elle faite avec des *sockets* ou les `RPC` ?

3 Analysez la tolérance aux pannes et bugs de votre architecture.

4 Dans quels cas le parallélisme est-il intéressant ?

5 Serait-il possible que plusieurs personnes éditent le même document en même temps en voyant les rectifications apportées par les autres en temps réel sur le document affiché sur leur écran ?

Si oui, si votre architecture le supporte expliquez pourquoi si elle ne le supporte pas expliquez comment on pourrait le faire.

Si non, donnez des exemples montrant que cela n'est pas réalisable.

6 Imaginons que l'on ajoute un module abréviation qui lorsque l'on insère les caractères "Âc/s" le remplace par "client/serveur", ou bien un module sélectionnant un phrase complète si l'on triple-clique dessus, ou bien un module ajoutant des menus.

Ces modules n'étant pas prévues dans l'architecture initiale du traitement de texte. Votre architecture supporte-t-elle ces ajouts sans modification d'aucun programme? Si oui comment? Si non, alors proposez une extension de votre architecture et/ou de vos protocoles afin de supporter ce genre d'ajout sans aucune restriction. (et sans avoir à modifier des programmes existants)

7 Donnez des exemples montrant les avantages et inconvénients de votre architecture et qui n'ont pas déjà été détaillé.