Examen Client/Serveur, 1997/1998

Tout document autorisé

Argumentez vos réponses.

1. Pourquoi faire du client/serveur ?
2. Quels avantages/inconvénients y a-t-il à faire du client/serveur si le client et le serveur fonctionnent sur la même machine ?
3. Citez deux outils permettant de mettre en communication un client et un serveur.
4. Qu'est-ce que le client `telnet` utilise pour communiquer avec le serveur (citez un outil de la question 3.) ?
5. Qu'est-ce que le client `NFS` (le système d'exploitation) utilise pour communiquer avec le serveur (citez un outil de la question 3.) ?
6. Citez des applications fonctionnant en client/serveur en spécifiant pour chacune si vous les connaissez :
   - Des noms de clients.
   - Des noms de serveurs.
   - Le nom du protocole.
   - La méthode de communication utilisée parmi celles indiquées dans la question 3.
7. Soit une application découpée en un client et un serveur, quel est **généralement** le rôle du client et celui du serveur.
8. Dans le cas de l'aide interactive de `NanoEdit` qui s'affiche dans un fenêtre `Mosaic`. Qui est le client et qui est le serveur ?
9. Quand un processus détruit un fichier sur le disque, ce n'est pas lui qui le fait, il demande au système d'exploitation de la machine de le faire. Qui est client et qui est serveur ? Donnez d'autre exemples avec ces même deux "acteurs".
10. Qu'est-ce qui doit être standardisé (ou au moins connu) afin que tout le monde puisse écrire d'autres versions des clients et des serveurs ?
11. Quand vous utilisez X Window, vous faite du client/serveur. Quel est le rôle du serveur X Window ? Citez des clients directs du serveur X ainsi que des clients indirects (A client de B, B client du serveur X).
12. Partant de `NanoEdit`, citez les serveurs utilisés par celui-ci, puis les serveurs utilisés par les serveurs précédents, ainsi de suite. Présentez ceci sous forme d'un graphe orienté que vous détaillerez le plus possible.
13. Toutes les applications doivent être considérées comme serveuses vis-à-vis de l'être humain qui les utilise. À votre avis, peut-on considérer que toute application peut-être serveuse vis-à-vis d'une autre application ? Si non : donnez des contre-exemples, si oui : pourquoi ?
14. Soit un client qui communique en utilisant des *sockets* avec un serveur sans crypter les messages échangés. Ne disposant pas de la possibilité de modifier les programmes client et serveur, comment faire pour crypter les messages ? Vous vous inspirerez du TP sur l'espion.
15. Il vaut mieux un serveur qui rend 10 services ou bien 10 serveurs rendant chacun un service ? Quels sont les avantages et inconvénients de ces deux choix ? Proposez un découpage du serveur «système d'exploitation» en plusieurs serveurs séparés.
16. Les CPU (processeurs) des machines (ou bien les langages de programmation) ne codent pas les entiers (ou d'autres types d'information) de la même façon. Comment faire pour que le clients et le serveur puissent dialoguer ? Indiquez les deux solutions les plus courantes ainsi que leurs avantages et inconvénients.
17. Qu'est-ce qu'un *proxy* ?
18. Pourquoi utilise t-on des *proxy* ?
19. Quand un serveur tombe en panne, il faut que les clients s'adressent a un autre serveur. Proposez différentes solutions afin que ce changement se fasse de manière transparente (sans intervention humaine)
20. Imaginons que vous ayez un réseau dans votre maison. Que les lampes et les interrupteurs soient des "machines" avec une adresse et puissent faire tourner des programmes. Qui est le client et le serveur ? Pensez-vous que l'on pourrait inverser les rôles ? Finalement, par quoi pourrait-on remplacer les noms client et serveur ?
21. Qu'auriez-vous aimé voir de plus dans le cours architecture client/serveur ?