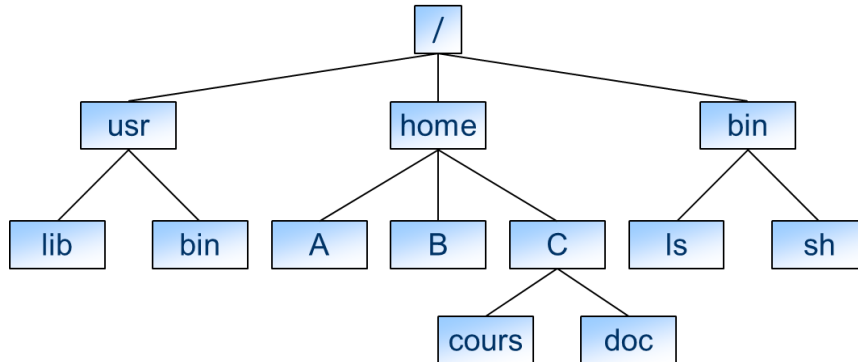


Exercice 1 – Montage

Soit le volume principal suivant :



1) Le dossier `/home/A` existe-t-il ?

Oui

2) Le dossier `/home/C/cours` existe-t-il ?

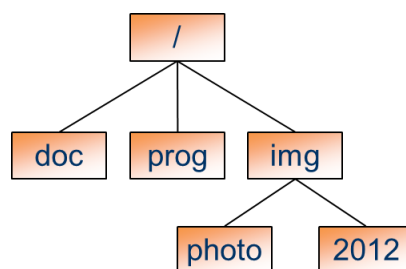
Oui

3) Le dossier `/home/B/cours` existe-t-il ?

Non

Autre exemple, depuis `/home/B`, se rendre dans le dossier `/home/C/cours` :

```
cd /home/B  
cd ../C/cours  
cd ~  
cd ~A  
cd -
```



4) On souhaite maintenant monter le volume ci-dessus contenu sur une clé USB dans le dossier `/home/A` du premier volume. Indiquer les commandes pour réaliser cette opération.

```
mount /dev/sdb1 /home/A  
umount /home/A
```

```
cat /etc/fstab
```

5) Comment accéder au dossier `doc` du nouveau volume ?

`cd /home/A/doc`

- 6) Comment accéder au dossier photo du nouveau volume ?

`cd /home/A/img/photo`
`cd ~A/img/photo`

Exercice 2 – NFS

- 1) NFS est un système de fichiers répartis "sans état". Rappelez les principes essentiels du fonctionnement "sans état" en particulier comment sont exécutées les requêtes successives lors d'une requête de lecture en mode séquentiel.

« sans état » : on ne conserve pas d'informations venant du client sur le serveur (fichier ouvert, etc...).

« sans état » => mode synchrone

L'ouverture du fichier se traduit par une requête bloquante entre le client et le serveur.

La lecture d'un bloc/morceau du fichier se traduit également par une requête bloquante entre le client et le serveur.

Et ces deux opérations se réaliseront sur le serveur de façon indépendante avec ouverture/fermeture du fichier à chaque fois.

- 2) Le mode de fonctionnement "sans état" autorise une technique simple de traitement des pannes du serveur au niveau du RPC. Quelle est-elle ? Expliquez pourquoi elle est tout à fait adaptée au mode "sans état".

Le fait que le mode « sans état » entraîne le mode « synchrone » oblige l'attente bloquante sur chacune des opérations entre le client et le serveur. Donc, en cas de pannes :

- *soit la requête du client est complètement terminée et dans ce cas-là, on n'a pas de problème ;*
- *soit la requête du client n'était pas complètement traitée au niveau du serveur, alors le client ne recevra pas d'acquiescement pour la requête signifiant une erreur.*

- 3) Soit un client A lisant le fichier /mnt/nfs/a 3 fois de suites, puis un client B accédant également à ce fichier, puis le client A y accédant une quatrième fois.

- a) Décrire les opérations nécessaires sans un cache NFS client.

Sans cache :

Client A veut lire le fichier A

Le serveur ouvre le fichier A, lit le fichier A, le ferme et envoie le résultat au client.

Client A veut lire le fichier A

Le serveur ouvre le fichier A, lit le fichier A, le ferme et envoie le résultat au client.

Client A veut lire le fichier A

Le serveur ouvre le fichier A, lit le fichier A, le ferme et envoie le résultat au client.

Client B veut lire le fichier A

Le serveur ouvre le fichier A, lit le fichier A, le ferme et envoie le résultat au client.

Client A veut lire le fichier A

Le serveur ouvre le fichier A, lit le fichier A, le ferme et envoie le résultat au client.

- b) Décrire les opérations nécessaires avec un cache NFS client.

Avec cache :

Client A veut lire le fichier A

Le serveur ouvre le fichier A, lit le fichier A, le ferme et envoie le résultat au client.

Client A veut lire le fichier A

Le serveur indique au client que le fichier n'a pas été modifié. Le client le récupère donc depuis son cache.

Client A veut lire le fichier A

Le serveur indique au client que le fichier n'a pas été modifié. Le client le récupère donc depuis son cache.

Client B veut lire le fichier A

Le serveur ouvre le fichier A, lit le fichier A, le ferme et envoie le résultat au client.

Client A veut lire le fichier A

Le serveur indique au client que le fichier n'a pas été modifié. Le client le récupère donc depuis son cache. (attention, si le client B a fait une modification sur le fichier A, alors, on n'utilise pas le cache).