Systèmes et Réseaux
Devoir 1 : Correction

Licence Informatique à distance
B.Herrmann - G.Laville

Exercice 1 : Structure physique

Question 1.1 : Collecte des numéros

Pour établir les numéros d'inode à utiliser dans le schéma, nous allons devoir utiliser plusieurs fois la commande ls de la manière suivante :

```
ls -ai /etc
```

1. L'option -a indique à ls de lister tous les fichiers du répertoire, y compris les fichiers cachés
2. L'option -i active l'affichage des numéros d'inode

Dans la partie i-Liste (liste des inodes) de la figure 1, nous avons l'inode 02 qui correspond à "root".

La partie donnée de cette inode 02 se décompose ainsi :
– son propre inode 02 (.)
– l'inode de son père (..) est également 02 : nous en déduisons que 02 est le répertoire racine du système (il est son propre père)
– l'inode 481 qui correspond au répertoire "etc"
L'inode 481 (répertoire "etc") dans la liste des inodes se décompose ainsi :
– son propre inode 481 (.)
– l'inode de son père (..) est 02
– l'inode 792 est sous cette arborescence et s'appelle "bash.d"
– l'inode 2900 est sous cette arborescence et s'appelle "hosts"
L'inode 2900 correspond à un bloc de données dans la liste des inodes : /etc/hosts est donc un fichier normal.

L'inode 792 référence un bloc logique de répertoire : On reprend donc la procédure décrite ci-dessus, et on trouve l'inode 5860, qui correspond à un fichier normal, /etc/bash.d/java.sh, constitué de deux blocs.

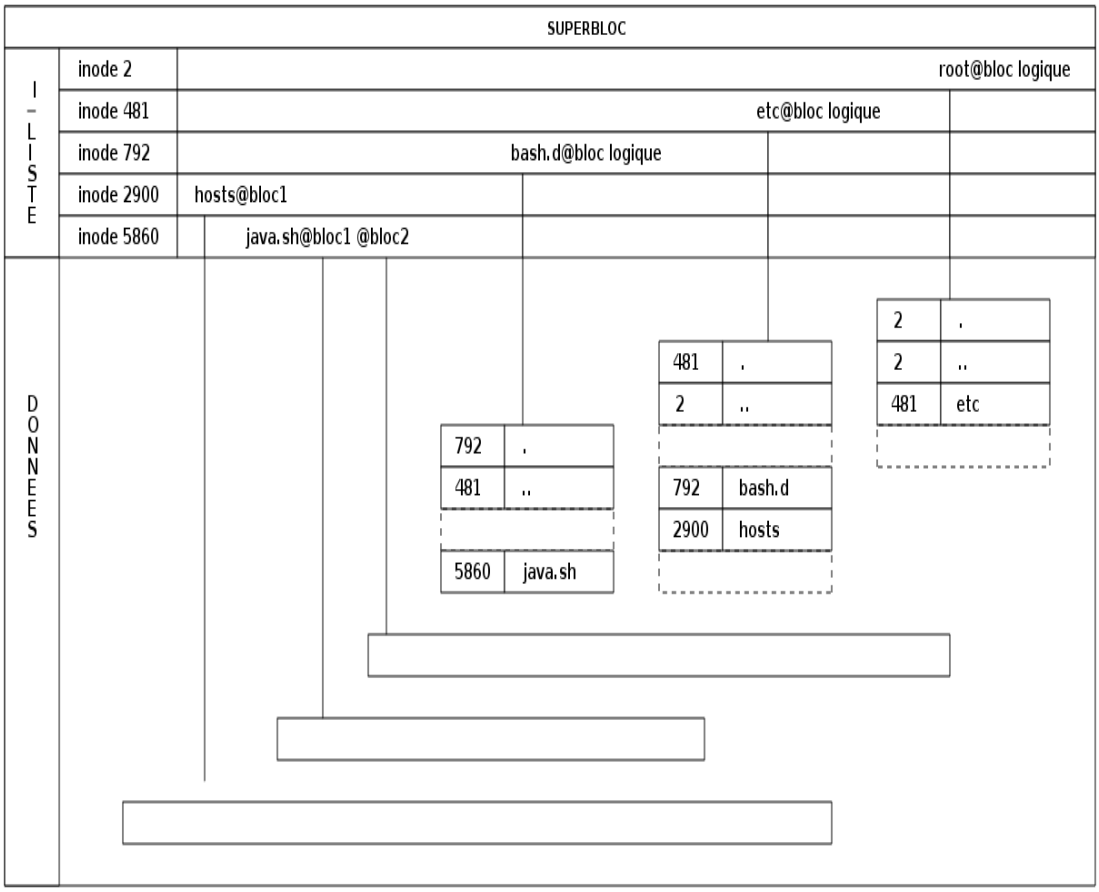

FIGURE 1 – Représentation physique du système de fichiers

Exercice 2 : Arborescence de fichiers

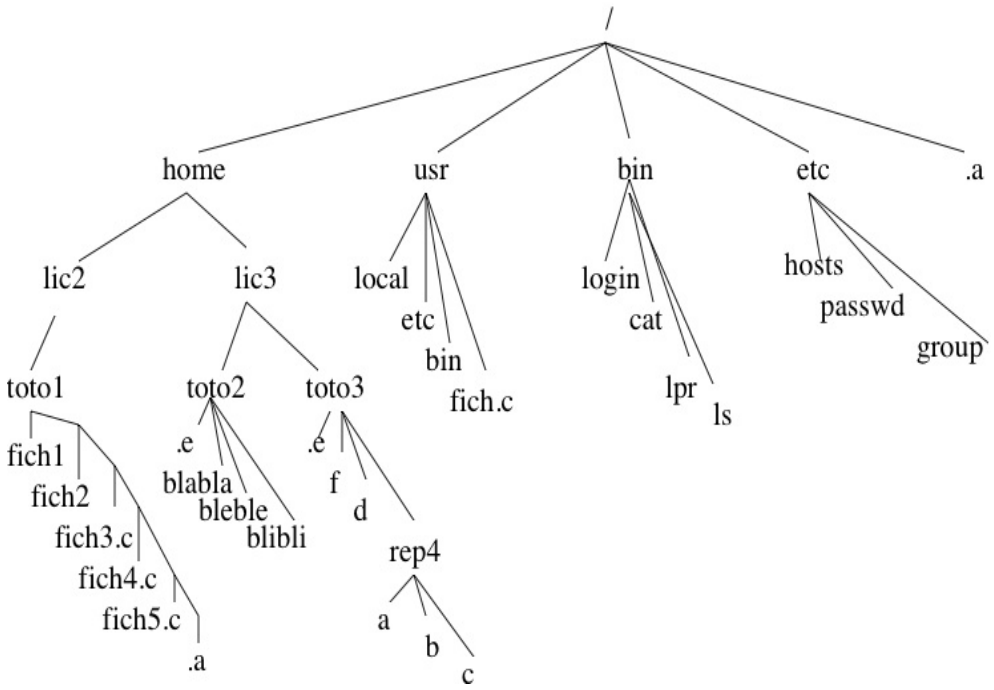

FIGURE 2 – Arborescence de fichiers

Question 2.1 : Dans l'arborescence précédente (figure 2) donner le résultat des commandes suivantes :

1. Depuis le répertoire *lic*3,
   `ls` : `toto2, toto3`
   `ls .` : `toto2, toto3`
   `ls ..` : `lic2, lic3`
   `ls toto3` : `f, d, rep4`
   `ls -a` : `., .., toto2, toto3`
   `ls -a .` : `., .., toto2, toto3`
   `ls -a ..` : `., .., lic2, lic3`
   `ls -a toto3` : `., .., .e, f, d, rep4`
2. Depuis *toto*1 donner deux manières de lister le contenu de *toto*2.
   `ls /home/lic3/toto2`
   `ls ../../lic3/toto2`
   `ls ~toto2`
3. Pour l'utilisateur *toto*2 :
   `ls ~` : `blabla, bleble, blibli`
   `ls ~toto1` : `fich1, fich2, fich3.c, fich4.c, fich5.c`
   `ls ~toto3/rep4` : `a, b, c`

Remarque : *les utilisateurs* `toto1`, `toto2` *et* `toto3` *doivent exister sur votre machine pour pouvoir tester les commandes utilisant le raccourci ~ pour désigner leurs répertoires d'accueil.*

Question 2.2 : Donner les commandes nécessaires pour réaliser :

1. si le répertoire courant est toto2, aller dans usr
   `cd ../../../usr ou cd /usr`
2. pour toto3, recopier fich2, sous rep4. Donner deux solutions.
   ```
   cp /home/lic2/toto1/fich2 rep4
   cp ~toto1/fich2 rep4
   cp ../../lic2/toto1/fich2 rep4
   ```
3. pour toto3, déplacer f sous rep4
   `mv f rep4`
4. pour toto3, créer le répertoire toto4 sous lic3
   ```
   mkdir ../toto4
   mkdir /home/lic3/toto4
   ```
5. pour toto3, effacer le répertoire rep4
   ```
   rm -r rep4
   rmdir rep4 # impossible si le repertoire n'a pas été préalablement vidé :
   rm rep4/*; rmdir rep4
   ```

Exercice 3 : Noms de fichiers génériques

Lors de l'évaluation d'une commande les caractères spéciaux sont évalués en premier par le shell, puis la commande est passée au système.

Question 3.1 : Donner le résultat des commandes suivantes pour l'utilisateur toto1 :

1. `ls .*` : le shell remplace *, cela donne .a . ..
   et applique la commande ls sur ce résultat (ex ls .):
   ```
   .a
   .  : fich1 fich2 fich3.c fich4.c fich5.c
   .. : toto1
   ```

   `ls *` : `fich1 fich2 fich3.c fich4.c fich5.c`

   `ls *.*` : `fich3.c fich4.c fich5.c`

   `ls /*` : le shell remplace l'étoile et applique la commande ls dessus:
   ```
   /home: lic2 lic3
   /usr: local etc bin fich.c
   /bin: login cat lpr ls
   /etc: hosts passwd group
   ```

   `ls ~toto2/* : toto2/blabla toto2/bleble toto2/blibli`

   Remarque : *dans * et *.*, on n'a pas les fichiers cachés commençant par . car par défaut le caractère étoile ne prend pas les fichiers cachés et la commande ls ne les affiche pas par défaut.*

2. `echo *  : fich1 fich2 fich3.c fich4.c fich5.c`

   Remarque : *Le shell remplace l'étoile par tous les noms fichiers et répertoires non cachés présents dans le répertoire courant, puis applique la commande echo sur ce résultat.*

   ```
   echo \* : *
   echo /  : /
   echo /* :/home /usr /bin /etc
   ```

3. `ls fich?.c: fich3.c fich4. fich5.c`

Question 3.2 : Donner une commande qui :

1. liste les commandes commençant par "l" sous /bin :
   `ls /bin/l*`
2. liste les commandes qui ne commencent pas par "l" sous /usr :
   `ls /usr/[!l]*`
3. les fichiers fich de 1 à 4 sous toto1 :
   `ls ~toto1/fich[1-4]*`
   `si on ne met pas l'étoile, on obtient seulement fich1 et fich2`
4. les fichiers fich sans le 3 sous toto1
   `ls ~toto1/fich[!3]*`
5. détruit les fichiers fich3.c, fich4.c, fich5.c :
   `rm ~toto1/fich[3-5].c`

Exercice 4 : Droits d'accès

Question 4.1 : Remarque : *Se pose un problème de terminologie avec cet exercice. Quand on demande de "donner" ou "attribuer" des droits, nous entendons que ces droits remplacent les anciens. Dans ce cas, on utilise la commande* `chmod` *avec un signe = ou l'octal. Quand on ajoute ou supprime des droits, on utilise la commande chmod avec les signes + ou -, on ne peut pas utiliser l'octal sauf si l'on connait les droits initiaux.*

Pour l'utilisateur *toto*2, donner la commande (avec rwx puis en octal si possible) qui :

1. donne le droit de lecture et écriture à tous sur blabla,
   `chmod a=rw blabla ou ou chmod =rw blabla chmod 666 blabla`
2. donne le droit de lecture et exécution à l'utilisateur et au groupe, rien pour les autres sur bleble,
   `chmod ug=rx,o= bleble ou chmod u=rx,g=u,o= bleble chmod 550 bleble`
3. donne lecture/écriture/exécution pour l'utilisateur, lecture/exécution pour le groupe et exécution pour les autres sur blibli,
   `chmod u=rwx,g=rx,o=x bliblbi ou chmod 751 blibli`
4. ajoute le droit de lecture aux autres sur blibli,
   `chmod o+r blibli`

5. supprime lecture/exécution au groupe sur bleble,
   `chmod g-rx bleble`
6. supprime lecture/écriture au groupe et les autres sur blabla.
   `chmod og-rw blabla`

Question 4.2 : Donner le masque de création pour que les fichiers soient créés par défaut :

1. uniquement avec le droit de lecture à l'utilisateur et rien pour le reste.
   `umask 266 (010 110 110) les permissions d'exécution sont supprimées à postériori pour les fichiers`
   `c'est le complément à 1 de 511 (101 001 001) ou r-x--x--x`

   `ou on peut aussi donner:`
   `umask 377 (011 111 111) fonctionne pour les fichiers mais ne donne pas le droit d'exécution pour les répertoires.`
   `C'est le complément à 1 de 400 (100 000 000) ou r--------`
2. avec un droit de lecture à tous et un droit d´écriture à l'utilisateur.
   `umask 022 (000 010 010) complément à 1 de 755 (111 101 101) ou rwxr-xr-x juste pour les fichiers`
   `umask 133 (001 011 011) complémént à 1 de 644 (110 100 100) ou rw-r--r--`

Question 4.3 : Lorsque l'on essaie de supprimer le fichier /etc/passwd, on a un message d'erreur En effet,

```
ls -l /etc
```

nous montre que ce répertoire n'a des droits d'écriture que pour son propriétaire, et que celui-ci est root. En effet, seules les permissions sur le répertoire parent interviennent lorsque l'on veut supprimer un fichier.

Question 4.4 : Droits à positionner :

1. créer un petit fichier de texte qui soit lisible par tout le monde, mais pas modifiable (même pas par vous).
   Il s'agit d'un fichier, pour le lire on a besoin uniquement des droits en lecture donc : r–r–r–
   `chmod 444 monfic`
2. créer un répertoire nommé topsecret, dont le contenu est visible uniquement par l'utilisateur .
   On attribue les droits de lecture à l'utilisateur , s'il veut lister le contenu du répertoire de façon détaillée, on doit ajouter les droits d'exécution
   `chmod 400 topsecret`
   `chmod 500 topsecret`

3. créer un répertoire nommé "petitsecret" tel que "knock" ne puisse pas lister son contenu mais puisse lire les fichiers qui y sont placés. *"knock" est un utilisateur quelconque, il fait partie de other. Pour pouvoir lire les fichiers placés dans le répertoire "petitsecret", par ex avec la commande "cat", il doit avoir les droits en exécution sur ce répertoire, les droits de lecture sur les fichiers qu'ils veut lire. On ne veut pas qu'il puisse lister le contenu du répertoire (commande ls), il n'a donc pas de droit de lecture sur celui-ci. On ne dit rien en ce qui concerne les droits du groupe sur ce répertoire.*
   `chmod 711 petitsecret`

Question 4.5 :

1. Dans quelles conditions l'utilisateur *toto*3 peut-il recopier le fichier f dans le répertoire *totot*2 ? Il doit
   - avoir le droit d'écriture (pour pouvoir créer dans le répertoire) et d'exécution (pour avoir accès au répertoire) sur `toto2`. Pour cela, ces droits sont attribués au groupe si `toto3` fait partie du même groupe que `toto2` ou alors à `other` sur le répertoire `toto2`.
   - avoir le droit de lecture sur `f` (lecture pour l'utilisateur sur le fichier `f`)
2. L'utilisateur *toto*3 peut-il effacer le fichier $fich1$?
   - L'utilisateur `toto3` doit avoir les droits d'exécution (accès pour pouvoir vérifier les droits) et d'écriture (pour la suppression) sur le répertoire `toto1` pour pouvoir effacer `fich1`. Ces droits sont attribués à `other` sur le répertoire `toto1`. Il peut y avoir création d'un nouveau groupe pour ne pas donner ces droits à `other`.
   - Aucun droit n'est requis sur le fichier fich1.

Exercice 5 : Commandes de sélection

Le fichier */etc/passwd* contient la description des utilisateurs. On suppose que le format du fichier est le suivant :

```
login:passwd:iud:gid:nom prenom:home_path:shell
```

Question 5.1 : Donner :

1. la liste des logins,
   `cut -f1 -d: /etc/passwd`
2. les trois premiers caractères de chaque ligne,
   `cut -c1-3 /etc/passwd`
3. le nombre d'utilisateurs,
   `wc -l /etc/passwd`
4. la liste des utilisateurs (nom prenom),
   `cut -d: -f5 /etc/passwd`

Question 5.2 : Donner une commande qui :

1. recherche blabla dans le fichier fich,
   `grep blabla fich`
2. recherche blabla dans tous les fichiers du répertoire,
   `grep blabla *`
3. affiche le contenu des fichiers dont le nom ne commence pas par une lettre.
   `cat [!a-zA-Z]*`