

# TD séances n° 5

## Gestion avancée de Fichiers sous Unix

### Exercices

## 1 Permissions sur les fichiers

### Exercice n°1:

1. Créez un répertoire `Linux` et déplacez-vous dans celui-ci  
`mkdir Linux`  
`cd Linux`
2. Créez le fichier vide `mon_fichier`, et examinez ensuite ses permissions.  
`touch mon_fichier`  
`ls -l mon_fichier`
3. Pour chacun des exercices suivants, donnez la commande `chmod` correspondante avec le changement de permissions en symbolique et en numérique. Donnez successivement au fichier les droits nécessaires pour que vous puissiez :
  - a. Lire, modifier et exécuter votre fichier  
`chmod u+rwX mon_fichier`
  - b. Lire, modifier mais ne pas exécuter votre fichier  
`chmod u-x mon_fichier`
  - c. Lire mais ne pas modifier ou exécuter votre fichier  
`chmod u-w mon_fichier`
4. Accordez maintenant toutes les permissions au propriétaire et la lecture seulement pour le groupe et rien pour les autres.  
`chmod 740 mon_fichier`
5. Maintenant tentez de consulter le fichier `mon_fichier` créé par votre voisin et testez ce que vous pouvez faire sur ce fichier. Expliquez.  

**Vous ne pouvez pas consulter le contenu de ce fichier, mais vous pouvez le voir dans la liste des fichiers si vous listez le contenu du répertoire.**
6. Positionnez les permissions nécessaires pour qu'un utilisateur de votre groupe puisse lire, modifier mais ne pas supprimer votre fichier. Que faut-il faire pour pouvoir supprimer le fichier ?  
`chmod g+rw mon_fichier`  
`chmod g-w .`  

**Pour pouvoir supprimer le fichier, il faut les droits d'écriture sur le répertoire (les droits d'écriture sur le fichier ne permettent que de modifier son contenu et pas de l'effacer)**

## 2 Liens physiques et symboliques

### Exercice n°2:

1. Créez dans votre répertoire `~` un répertoire `tmp` qui contient un fichier `bidon`. A l'aide de `gedit`, ajoutez une ligne de texte dans le fichier `bidon`.  
`cd ~`  
`mkdir tmp`  
`touch tmp/bidon`  
`gedit tmp/bidon` (mettre une ligne de texte dans le fichier et quitter gedit)
2. Dans votre home directory (`~`), créez un lien physique appelé `dhuile` vers le fichier `tmp/bidon`. Comparez les contenus de `tmp/bidon` et de `~/dhuile`. Que contient `dhuile` ?  
`ln tmp/bidon ./dhuile`  

**Le contenu de `dhuile` est strictement identique à ce que l'on a mis dans `tmp/bidon`.**

## TD séances n° 5

### Gestion avancée de Fichiers sous Unix

3. Notez les droits que vous avez actuellement sur le fichier ~/dhuile. Modifiez les droits sur le fichier tmp/bidon pour avoir les permissions suivantes rw-r-----. Quels sont les droits d'accès sur le fichier ~/dhuile ?

Les droits après création du fichier tmp/bidon sont : -rw-rw-r-- et après création du lien physique, on a les mêmes sur le fichier dhuille.

```
chmod 640 tmp/bidon
```

Les droits sur le fichier dhuille sont les mêmes que sur le fichier tmp/bidon après la modification : rw-r-----. Donc les deux fichiers ont toujours les mêmes droits.

4. Supprimez le fichier tmp/bidon puis consultez le contenu du fichier dhuille. Que constatez-vous ?

```
rm tmp/bidon
```

Le fichier dhuille a toujours le contenu qui avait été saisi dans le fichier tmp/bidon

5. Après avoir effacé le fichier dhuille, refaites les questions 1, 2 et 3 de cet exercice, mais au lieu de faire un lien physique, faites un lien symbolique.

```
cd ~
```

```
mkdir tmp
```

```
touch tmp/bidon
```

```
gedit tmp/bidon (mettre une ligne de texte dans le fichier et quitter gedit)
```

```
ln -s tmp/bidon ./dhuille
```

Le contenu de dhuille est strictement identique à ce que l'on a mis dans bidon.

Les droits après création du fichier tmp/bidon sont : -rw-rw-r--. Les droits sur le lien symbolique dhuille sont : lrwxrwxrwx

```
chmod 640 tmp/bidon
```

Les droits sur le fichier dhuille ne sont pas modifiés et toujours égaux à lrwxrwxrwx après la modification des droits sur le fichier tmp/bidon : rw-r-----. Donc une modification des droits sur le fichier ne modifie pas les droits sur le lien symbolique. Le lien symbolique n'a pas ses propres droits et ce sont ceux du fichier pointé qui comptent.

```
rm tmp/bidon
```

Lorsque l'on essaye de consulter le contenu du fichier dhuille, on a une erreur (le fichier n'existe plus)

6. Quelles sont les différences entre les liens physiques et les liens symboliques ?

Dans le cas d'un lien physique, on donne un deuxième nom au fichier et tous les deux pointent vers le contenu. Si je n'efface que l'un des deux fichiers, je peux toujours consulter le contenu.

Dans le cas d'un lien symbolique, je ne crée qu'un nom qui pointe sur un nom existant. Si je détruit le premier nom de fichier, le lien ne pointe plus sur rien.

7. Faites un lien physique de nom recherche dans /tmp sur le fichier /usr/bin/find. Que se passe-t-il ? En déduire dans quel cas on ne peut pas faire de lien physique ? Que faut-il faire alors ?

```
ln /usr/bin/find /tmp/recherche
```

On a pas les droits pour le faire. Dans ce cas, on ne peut pas faire de lien physique donc on fait un lien symbolique. Mais attention, ce n'est pas exactement la même chose.

## 3 Alias

### Exercice n°3:

Faire un alias qui permet de voir les fichiers cachés et d'afficher les fichiers en couleur.

```
alias ll='ls -l --color'
```

## TD séances n° 5

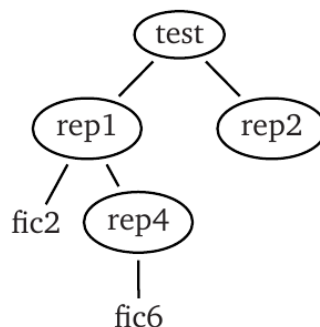
# Gestion avancée de Fichiers sous Unix

## Synthèse

Nous allons explorer un peu plus en détail les possibilités du système de fichier UNIX à travers le concept de lien et la notion de numéro d'i-nœud.

### Exercice A :

- Pour commencer, créez l'arborescence suivante dans votre dossier *homedir*.



```

cd ~
mkdir -p test/rep1/rep4 test/rep2
touch test/rep1/fic2 test/rep1/rep4/fic6
  
```

- Donnez trois façons de désigner le fichier *fic6* depuis votre *homedir*.  
`/home/user/test/rep1/rep4/fic6`  
`~/test/rep1/rep4/fic6`  
`test/rep1/rep4/fic6`
- À l'aide d'un éditeur de texte, écrivez *Il fait beau aujourd'hui !* dans le fichier *fic6*.  
`gedit test/rep1/rep4/fic6`
- À l'aide de la commande *cat*, affichez le contenu du fichier *fic6* depuis votre répertoire *rep2*.  
`cd test/rep2`  
`cat ../rep1/rep4/fic6`
- L'option *-l* de la commande *ls* permet entre d'autre d'observer les droits d'un fichier/répertoire. À quoi correspondent les autres informations que l'on obtient grâce à cette commande ?  
`ls -l ~/test/rep1`  
 On obtient les informations sur le nombre de lien physique vers un nom, le propriétaire, le groupe, la taille en octets, la date de création et le nom de fichier.

### Exercice B :

- Comme expliqué dans le cours, la commande « *ln* » sert à créer des *liens*. Utilisez-la pour créer un lien **physique** du fichier *fic6* dans *test* sous le nom de *lpfic6*.  
`cd ~/test`  
`ln rep1/rep4/fic6 lpfic6`
- Modifiez le contenu du fichier *lpfic6*. Que constatez-vous pour le fichier *fic6* ? Réciproquement, modifiez *fic6*, lisez *lpfic6*. Concluez.  
`gedit lpfic6` (ajout de texte)  
 Le fichier *fic6* contient aussi le texte ajouté à *lpfic6*. Idem si je modifie *fic6*, *lpfic6* contient les mêmes informations
- Modifiez les droits d'accès au fichier *fic6* pour les membres du groupe. Que constatez-vous pour le fichier *lpfic6* ? Pouvez-vous avancer une explication ?  
`chmod 640 rep1/rep4/fic6`

## TD séances n° 5

### Gestion avancée de Fichiers sous Unix

```
ls -l lpfic6
```

Les droits du fichier `lpfic6` sont aussi `640`. Les droits d'accès à un fichier ne dépendent donc pas du nom du fichier mais sont bien associés au contenu lui même du fichier.

- La commande « `ln` » peut aussi créer des liens *symboliques* avec l'option `-s`. Créez un lien symbolique du fichier `fic6` dans `test` que vous appelez `lsfic6`.

```
ln -s rep1/rep4/fic6 lsfic6
```

- Regardez toutes les informations concernant les fichiers `lpfic6` et `lsfic6`. Quelles différences notez-vous ?

Le fichier `lpfic6` est un fichier normal, le premier caractère sur la ligne est un « `-` ». Le fichier `lsfic6` est un lien symbolique, le premier caractère de la ligne est un « `l` ».

- Essayez de modifier les droits d'accès au fichier `lsfic6`. pour les mettre à tous les droits pour les utilisateurs et le groupe, mais rien pour les autres. Que constatez-vous ?

```
chmod 770 lsfic6
```

Les permissions du lien symbolique ne changent pas, mais les permissions sont changées sur le fichier pointé `fic6`. Comme `lpfic6` est un lien physique sur le fichier `fic6`, lui aussi prend les permissions appliquées.

- Modifiez les droits d'accès au répertoire `rep1` pour ne plus y avoir accès. Essayez d'afficher le contenu de `lpfic6` et `lsfic6`. Que constatez-vous ? Pouvez-vous avancer une explication ?

```
chmod 000 rep1
```

```
cat lpfic6 : affiche le contenu du fichier
```

```
cat lsfic6 : dit que l'on a pas les droits (accès impossible dans le répertoire pointé par le lien)
```

- Modifiez de nouveau les droits d'accès au répertoire `rep1` pour y avoir de nouveau accès. Déplacez le fichier `fic6` dans le répertoire `rep1`. Essayez d'afficher le contenu de `lpfic6` et `lsfic6`. Que constatez-vous ?

```
chmod 755 rep1 (rétabli les droits comme ils étaient avant)
```

```
mv rep1/rep4/fic6 rep1/
```

```
cat lpfic6 : affiche le contenu du fichier
```

```
cat lsfic6 : dit que le fichier n'existe pas
```

- Redéplacez le fichier `fic6` dans le répertoire `rep4`. Essayez à nouveau d'afficher les contenus de `lpfic6` et `lsfic6`. Supprimez le fichier `fic6` puis recommencez. Que constatez-vous ?

```
mv rep1/fic6 rep1/rep4/
```

```
cat lpfic6 : affiche le contenu du fichier
```

```
cat lsfic6 : affiche le contenu du fichier
```

```
rm rep1/rep4/fic6
```

```
cat lpfic6 : affiche le contenu du fichier (mais si on fait ls -l lpfic6, on a plus que 1 pour le compteur d'utilisation)
```

```
cat lsfic6 : dit que le fichier n'existe pas
```

Pour la suite de la synthèse d'exercices, recréez le fichier `fic6` dans le répertoire `rep4`.

```
touch rep1/rep4/fic6
```

#### Exercice C :

- Observons maintenant plus attentivement le répertoire `rep2`. Combien y a-t-il de liens sur ce répertoire ? À quoi correspondent-ils ?

```
cd ~/test
```

```
ls -la
```

```
drwxrwxr-x 2 epu epu 4096 sep. 12 2013 rep2
```

Il y a 2 liens physiques vers le répertoire `rep2`. Ces deux liens correspondent au nom `rep2` dans `test` et au répertoire `.` quand on est dans le répertoire `rep2`.

## TD séances n° 5

### Gestion avancée de Fichiers sous Unix

- Dans le répertoire `rep2`, créez un sous-répertoire `rep3`. Combien y a-t-il maintenant de liens sur le répertoire `rep2` ? Expliquez.

```
mkdir rep2/rep3
```

Il y a maintenant 3 liens physiques vers le répertoire `rep2` (on a rajouté le lien `..` depuis le répertoire `rep3`)

- Créez un lien physique `lprep4` du répertoire `rep4` dans le répertoire `test`. Que remarquez-vous ?

```
cd ~/test
```

```
ln rep1/rep4 lprep4
```

La création d'un lien physique sur un répertoire est impossible

Avec les implémentations existant actuellement, seul le super-utilisateur peut créer un lien matériel sur un répertoire, et encore, ce n'est pas toujours possible. Par contre, on peut tout à fait créer un lien symbolique sur un répertoire.

- Créez un lien symbolique `lsrep4` du répertoire `rep4` dans le répertoire `test`.

```
ln -s rep1/rep4 lsrep4
```

- Créez un lien symbolique `lslsrep4` du lien `lsrep4` dans le répertoire `rep2`.

```
cd rep2
```

```
ln -s ../lsrep4 ./lslsrep4
```

- Donnez trois manières différentes de se déplacer dans le répertoire `rep4` à partir du répertoire `test`.

```
cd ~/test/rep1/rep4
```

```
cd ~/test/lsrep4
```

```
cd ~/test/rep2/lslsrep4
```

- En utilisant successivement ces trois méthodes, déplacez vous dans le répertoire `rep4` puis remonter dans le répertoire parent à l'aide de la commande `cd ..`. Que remarquez-vous ?

```
cd ~/test/rep1/rep4 ; cd .. (on se retrouve dans le répertoire rep1)
```

```
cd ~/test/lsrep4 ; cd .. (on se retrouve dans le répertoire test)
```

```
cd ~/test/rep2/lslsrep4 ; cd .. (on se retrouve dans le répertoire rep2)
```

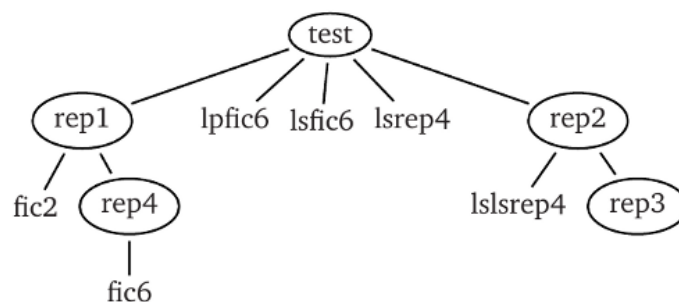
- Que se passe-t-il si on utilise la commande `ls -Ral` sur `lsrep4` ? et sur `lslsrep4` ?

```
ls -Ral ~/test/lsrep4/
```

```
ls -Ral ~/test/rep2/lslsrep4/
```

On obtient la même chose.

À la fin de cet exercice, nous avons maintenant l'arborescence suivante :



#### Exercice D :

- À l'aide de la commande `ls` munie de l'option appropriée, observez le numéro d'i-nœud du fichier `fic2`. Copiez le fichier `fic2` dans le répertoire `rep3`. Quel est son numéro d'i-nœud ?
- Changez le nom de ce dernier fichier, pour l'appeler `fic6`. Le numéro d'i-nœud change-t-il ?
- Comparez les numéros d'i-nœuds entre le fichier `fic6` du répertoire `rep4`, `lpfic6` et `lsfic6`. Que remarquez-vous ? Expliquez maintenant plus clairement les dernières questions de l'exercice B.

## TD séances n° 5

### Gestion avancée de Fichiers sous Unix

---

- Observez maintenant le numéro d'i-nœud de la racine et de votre répertoire personnel. Que remarquez-vous ?