

## Atelier 'Gérer les FS Ext2/3/4'

*man mkfs, man e2fsck, man tune2fs, man blkid, man fsck ... ☺  
man mount, man ln ☺*

1. Ajoutez un disque dur de 900 Mo à votre système et découpez-le en 3 partitions primaires de 300 Mo chacune.

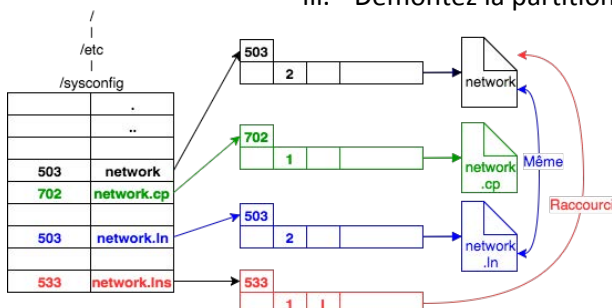
**Procédé: Voir Billet 'Installation'**

*Ici, le disque dur créé est supposé être géré par le device /dev/sdc...*

2. Listez la documentation des commandes de gestion des FS  
`#man -k ext2`
3. Créez un FS Ext2 en silence sur la 1<sup>ère</sup> partition avec des blocs de 2 Ko, en vérifiant la présence de mauvais blocs et en dimensionnant la table des i-nodes à 1000 i-nodes.  
`#mkfs -ext2 -q -b 2048 -c -N 1000 /dev/sdb1`
4. Créez un journal, le FS devient donc un Ext3  
`#tune2fs -j /dev/sdb1`
5. Visualisez les caractéristiques du FS.  
`#tune2fs -l /dev/sdb1 | more`
6. Montez le FS sans activer le journal.  
`#mount -t ext2 /dev/sdb1 /mnt`
7. Remontez le FS en tant que Ext3, on active donc la journalisation. `#umount /mnt`  
`#mount -t ext3 /dev/sdb1 /mnt`
8. Montez le FS Ext3 en tant que Ext4. `#umount /mnt`  
`#mount -t ext4 /dev/sdb1 /mnt`
9. Convertissez le FS Ext3 en Ext4. Mais il ne sera plus montable en Ext3.  
`#umount /mnt`  
`#tune2fs -O extents, ininit_bg, dir_index /dev/sdb1`  
`#e2fsck /dev/sdb1`
10. Les liens:

Créez:

- a. Une copie 'network.cp' du fichier /etc/sysconfig/network.  
`#cp /etc/sysconfig/network /etc/sysconfig/network.cp`
    - i. Listez les i-nodes relatifs à ces 2 fichiers  
`#ls -li /etc/sysconfig/network`
    - ii. Schématisez la structure interne de ces 2 entrées.
  - b. Un lien (dur) 'network.ln' sur le fichier /etc/sysconfig/network.  
`#cd /etc/sysconfig/`  
`#ln network network.ln`
    - i. Listez les i-nodes relatifs à ces 2 fichiers  
`#ls -li /etc/sysconfig/network`
    - ii. Schématisez la structure interne de ces 2 entrées.
  - c. Un lien symbolique 'network.lns' sur le fichier /etc/sysconfig/network.  
`#ln -s network network.lns`  
`#ls -li`
    - i. Listez les i-nodes relatifs à ces 2 fichiers
    - ii. Schématisez la structure interne de ces 2 entrées ainsi que le chemin parcouru dans le FS afin d'exécuter la commande "# cat network.lns".
- ci. Un lien symbolique '/mnt/grub/' vers le dossier /boot/grub.  
`#ln -s /mnt/grub/ /boot/grub`
    - i. Listez l' i-node relatif à cette nouvelle entrée.
    - ii. Schématisez la structure interne de cette entrée.
    - iii. Démontez la partition /boot et relistez l'i-node du lien symbolique.



## Différences entre liens durs et liens symboliques:

LIENS DURS	LIENS SYMBOLIQUES
Fait référence à un espace sur le disque dur.	Fait référence à une entrée dans un répertoire.
La suppression d'un des fichiers liés n'affectera pas l'autre	La suppression du fichier source entraînera un changement de comportement du fichier lié qui ne correspondra plus à un fichier valide et sera donc "cassé" ("broken").
Pour supprimer l'accès aux données du fichier, il faut supprimer tous ses liens (via la commande rm).	Pour supprimer l'accès aux données du fichier, il faut supprimer le fichier source (attention aux fichiers cassés !)

### Utilités

- Mise à disposition de fichiers à des utilisateurs sans avoir besoin de dupliquer ces données dans leur espace de travail.
- Dans ce cas, ces utilisateurs travaillent automatiquement sur la version actuelle des fichiers.
- Accès à des fichiers ou dossiers via plusieurs noms possibles:

Ex1.     /boot/grub/menu.lst                      ~         /boot/grub/grub.conf  
          /etc/grub.conf                         ~         /boot/grub/grub.conf

⇒ Ces quelques écritures sont équivalentes:

```
# cat /boot/grub/grub.conf
# cat /boot/grub/menu.lst
# cat /etc/grub.conf
```

Ex2.     /dev/cdrom                             ~         /dev/sr0

⇒ Ces quelques écritures sont équivalentes:

```
# mount /dev/cdrom /media
# mount /dev/sr0 /media
```

Ex3.     /etc/init.d                            ~         /etc/rc.d/init.d  
          /etc/rcx.d                           ~         /etc/rc.d/rcx.d  
          /etc/rc.d/rc3.d/ S10network       ~         /etc/rc.d/init.d/network

⇒ Ces quelques écritures sont équivalentes:

```
# cat /etc/rc.d/init.d/network
# cat /etc/init.d/network
# cat /etc/rc.d/rc3.d/S10network
# cat /etc/rc3.d/S10network
# cat /etc/rc.d/rc3.d/S10network
# cat /etc/rc3.d/S10network
```

etc...

### **Questions de réflexion:**

1. La commande `rm` permet, dans tous les cas, de supprimer l'accès aux données physiques d'un fichier. Vrai ou faux et pourquoi ?
  2. Après avoir créé un répertoire, son nombre de liens égal 2 et sa taille = 4096 bytes. Pourquoi ?
  3. Si le nombre de liens associé à un répertoire égal 4, qu'est-ce que cela signifie ?
  4. On ne peut pas créer un lien dur vers un autre FS. Pourquoi ?
  5. Pourquoi ne peut-on pas lier en dur des répertoires entre eux?
- 
1. *FAUX, si un lien dur a été fait en amont de la commande `rm`, nous pourrions toujours avoir accès aux données physiques du fichier.*
  2. *Le nombre de lien est égal à 2 car il possède déjà deux entrées à savoir "." et "..". Et sa taille est de 4096 bytes car c'est la plus petite unité d'allocation pour un bloc.*
  3. *Cela signifie que le répertoire possède les entrées "." et ".." ainsi que deux autres répertoires dans son arborescence.*
  4. *Un lien dur, c'est lorsque que 2 fichiers possèdent le même id dans la table des i-node. Chaque fs possède sa propre table d'i-node. Ce n'est donc pas possible de créer un lien dur entre leurs fichiers. Par contre un lien symbolique est possible*
  5. *C'est le système qui empêche ça car l'on souhaite éviter de se retrouver dans une boucle infini avec le père appelant le fils et le fils le père.*

## **PRINCIPALES COMMANDES MANIPULEES LORS DE CET ATELIER**

---

### **Principales commandes:**

`mkfs`, `blkid`, `tune2fs`, `mount`, `e2fsck`, `fsck`, `ln`