

Licence Professionnelle

Remise à Niveau

N°2 : Système de fichiers

Système de fichiers

- Droits
- Inodes
- Liens

Accès aux ressources

- Linux multi-utilisateurs => protéger l'accès aux ressources
- Pour chaque ressource un "**droit d'accès**" est fixé :
 - 3 types de permissions principaux :
 - lecture (**r**)
 - écriture (**w**)
 - exécution (**x**)
- Le "**mode d'accès**" définit les droits d'accès pour chaque type d'utilisateurs :
 - 3 types d'utilisateurs :
 - le propriétaire (user : **U**)
 - les personnes du groupe du propriétaire (group : **G**)
 - les autres (other : **O**)

Les droits d'accès aux fichiers

Le droit d'accès est interprété en fonction du **type de ressource** :

- Droit **r** (*lecture*):
 - répertoire : affichage de la liste des fichiers
 - fichier : affichage du contenu du fichier
- Droit **w** (*écriture*):
 - répertoire : modification du contenu du répertoire
 - fichier : modification du contenu du fichier
- Droit **x** (*exécution*):
 - répertoire : droit de traversée
 - fichier : droit d'exécution

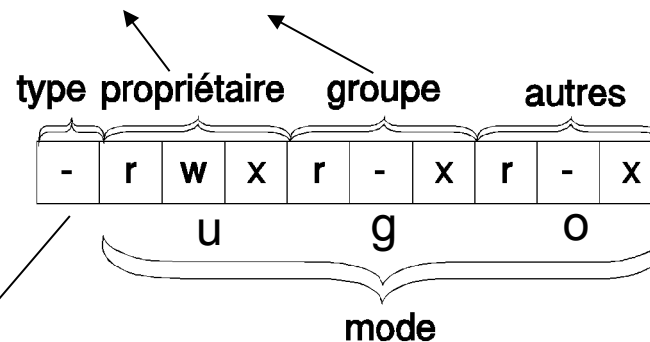
Remarque : protéger un fichier, c'est :

- 1) protéger son contenu : pas de droit w sur le fichier
- 2) interdire sa suppression : pas de droit w sur le répertoire d'appartenance

Afficher le mode d'accès

➤ Commande : **ls -l**

-rwxr-xr-x 1 util1 users 512 Jul 8 15:08 unFichier



➤ Types possibles

- fichier régulier
- d répertoire
- c, b périphérique (caractère ou bloc)
- l, s, p lien, socket, pipe

Droits spéciaux : s s t

1) Le droit « s » : droit d'endossement

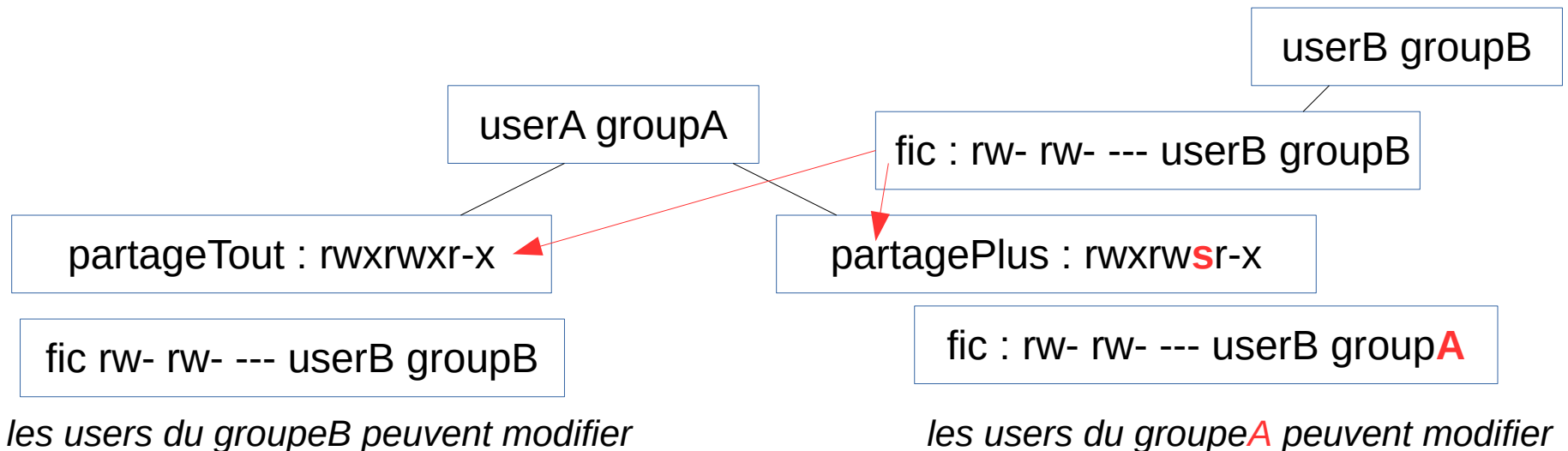
A) « s » pour un fichier exécutable

- Rappel : par défaut, un exécutable accède aux fichiers avec les droits de l'utilisateur l'ayant lancé
 - le droit **s** peut remplacer le droit **x** au niveau du propriétaire (u) et/ou du groupe (g) :
 - => l'utilisateur hérite alors des droits (droits effectifs) du propriétaire de l'exécutable durant son exécution
- Exemple : la commande passwd (/usr/bin)
 - pour permettre à un utilisateur de changer son mot de passe il faut accéder en écriture au fichier /etc/passwd (*réservé à root*)
 - la commande passwd s'exécute avec les droits root

Le droit « S »

B) « s » pour un répertoire (au dessus du droit *rwX*)

- « s » pour le **groupe**,
- signifie que les fichiers du répertoire seront créés comme appartenant au groupe du propriétaire du répertoire
- *et non comme par défaut au groupe du propriétaire du fichier*
- Exemple : céder les droits de modification à un autre groupe d'utilisateurs



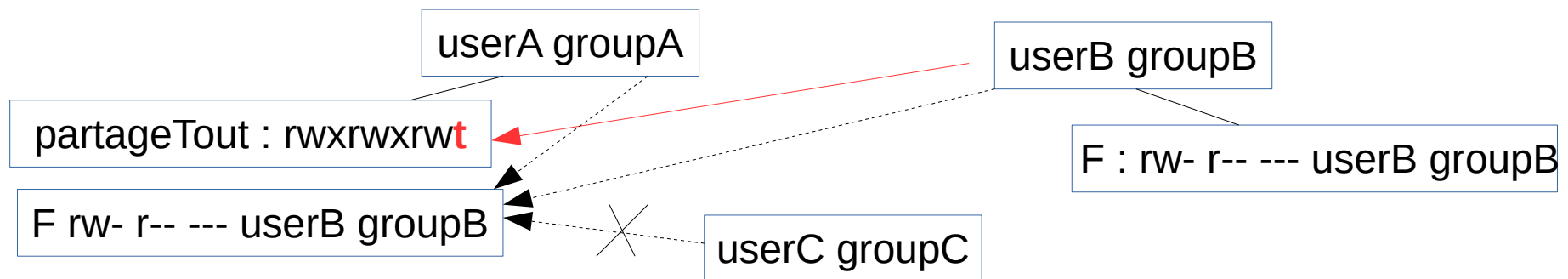
Le droit « **t** »

2) Le droit « **t** » : sticky bit (*au dessus du droit wx*)

- « **t** » pour un répertoire
- « **t** » pour les autres (other):
 - Seuls les propriétaires des fichiers peuvent les détruire
 - *sinon, par défaut, droit sur le répertoire <=> tous peuvent supprimer*

➤ Exemple d'usage : un répertoire partagé (ex: /tmp)

- tout le monde peut déposer, seul le propriétaire du fichier et celui du répertoire peuvent détruire



Rmq : s,s, et t apparaissent en majuscules si le droit x n'est pas dessous

Définition des droits d'accès

- Le **propriétaire** d'une ressource est le seul, *avec root*, autorisé à modifier son mode d'accès :
 - **chmod** : modifier le mode d'accès
 - fonctionne selon 2 modes : symbolique ou octal
- *root dispose en plus des commandes :*
 - **chown** : *Changer le propriétaire (et le groupe)*
 - **chgrp** : *Changer le groupe*

Mode d'accès : Notation symbolique

chmod *classes opérations permissions fichiers*

Classes	Opérations	Permissions
u : utilisateur g : groupe o : autres a : tous	= : affectation de droits - : suppression de droits + : ajout de droits	r : lecture w : écriture x : exécution s : endossement t : sticky bit

Exemple :

chmod **-R o+rx** ~ # lecture et exécution autorisées pour les autres

chmod **a+s , g=** *.sh # endossement pour les scripts

séparateur

aucun droit

Mode d'accès : Notation octale

chmod *nnnn fichiers*

- ▶ A chaque permission est associée une valeur :

valeur	permission
4	r
2	w
1	x
0	aucune

- ▶ Exemple : droit rw- \Leftrightarrow 6 ($4+2 \Leftrightarrow$ valeur octale 110)
- ▶ Les permissions doivent être précisées pour toutes les classes d'utilisateurs
 - ▶ Ex : *propriétaire rwx, groupe rw, autres r* \Rightarrow chmod 764 fichier
- ▶ Notation octale : pour les affectations *globales*
- ▶ Notation symbolique : pour les modifications

Droits spéciaux : Notation octale

- Droits spéciaux => 4^{ème} octet
 - Le 1^{er} chiffre représente les droits spéciaux (sst) :
 - 4 : setuid,
 - 2 : setgid,
 - 1 : sticky bit

- ▶ Exemple :
 - 4 5 5 5 => ?
 - 5 5 5 => r-x r-x r-x
 - 4 => 1 0 0 => s - -
 - => r-s r-x r-x

- ▶ Attention :
 - `chmod 4555 f ; chmod 2555 f ;` => `f : 6555 !!!`
 - => effacer les droits spéciaux en symbolique : `u-s,g-s,o-t`

Droits par défaut à la création

- Les droits **maximaux** :
 - droits accordés par défaut lors de leur création par linux
 - fichier : **666**
 - répertoire : **777**

- Attention :
 - cp : crée un fichier avec les mêmes droits que le fichier origine si le fichier cible n'existe pas sinon les droits de la cible ne sont pas changés
 - Exemple :


```
date >ladate (666)
chmod 600 ladate (600)
cp ladate lejour (600)
chmod 622 lejour (622)
cp lejour ladate (600)
```

Droits à la création

- La commande **umask**
 - Un masque de droits (*éventuellement nul*) est toujours appliqué sur les droits maximaux (*i.e. retire certains droits par rapport au maximum*)
 - La commande umask permet de manipuler le masque de droits
umask *valeur_du_masque_en_octal*
 - Par défaut la valeur du masque est 0022
 - 0000 0000 0010 0010 <=> --- --- -w- -w-
 - Remarques : Modifie pour
 - les prochaines créations (*pas les droits sur les ressources existantes*)
 - uniquement pour le shell courant

Exemple droits à la création

- umask 0026 (--- --- -w- rw-)

- touch fic (création : droits maximaux=666 (--- rw- rw- rw-))

droits maximaux	:	- rw- rw- rw- (666)
masque	:	- --- -w- rw- (026)
droits de fic	:	- rw- r-- --- (640)

- mkdir rep (répertoire : droits maximaux=777)

droits maximaux	:	- rwx rwx rwx (777)
masque	:	- --- -w- rw- (026)
droits de rep	:	- rwx r-x --x (751)

- Remarque : 6 6 6 - 1 1 1 = 6 6 6

Attributs étendus des fichiers

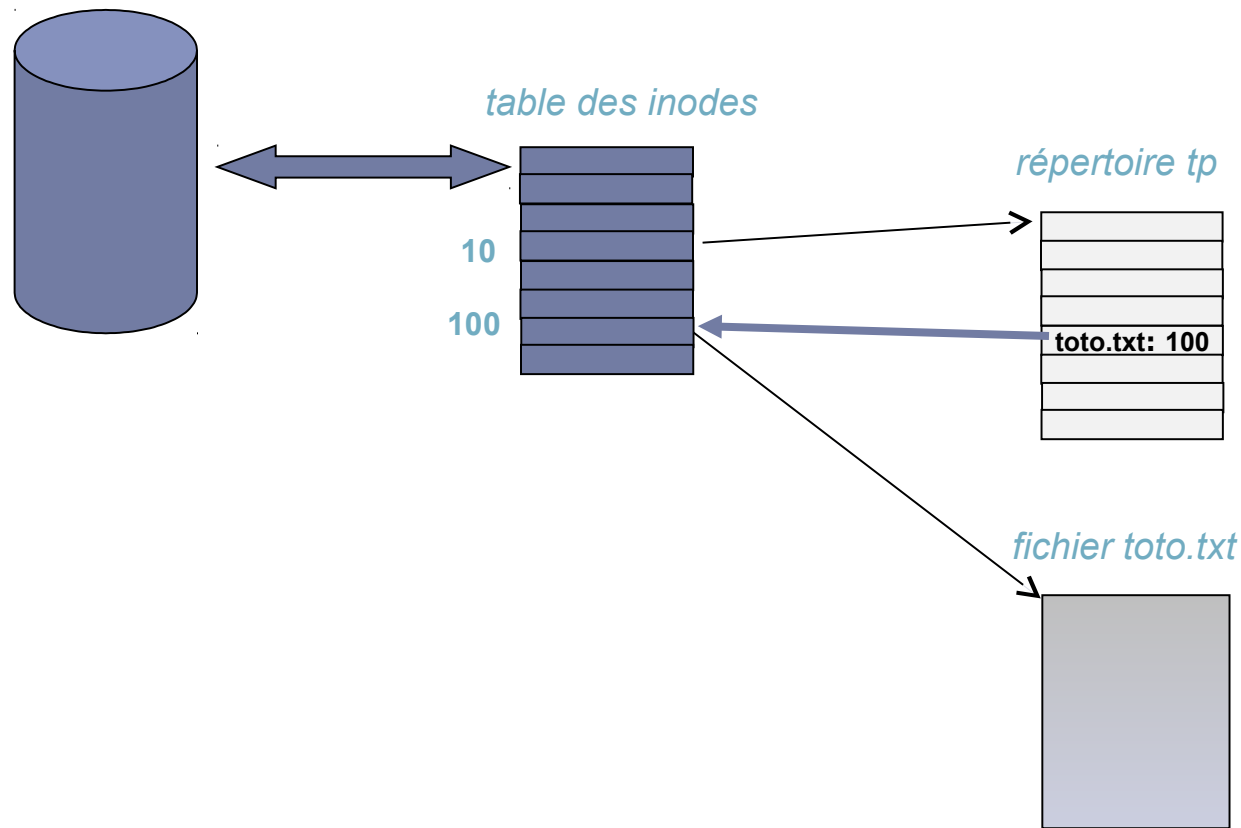
- Pour les systèmes de fichiers **ext**, indépendamment du mode d'accès, des attributs étendus peuvent être spécifiés pour les fichiers ordinaires et les répertoires
 - Exemples
 - **A** (no Access time) : pas maj de la date
 - **a** (append only) : (root) pas suppression de contenu
 - **i** (immutable) : (root) modif et suppression interdites (même à root)
 - **s** (secure deletion) : effacement sécurisé raz
 - **d** (nodump) : pas sauvegardé...
 - Commandes
 - **lsattr** : lister les attributs étendus d'un fichier
 - **chattr** : changer les attributs étendus d'un fichier
- ▶ Ces attributs sont ignorés lors des copies et déplacements

Les inodes

- A chaque fichier du système est associé une structure d'information, appelée **inode**, permettant la gestion par le système de fichiers.
- Il existe une et une seule **table des inodes** par système de fichiers (partition).
- L'inode est repérée par **son indice** dans cette table.
- La commande **ls -i** permet de lister les fichiers en affichant leur numéro d'inode.
- Un répertoire est un fichier particulier, qui contient une table qui associe à chaque nom de fichier, une entrée dans la table des inodes.

Table des inodes

dans un répertoire *tp* (*inode 10*)
se trouve un fichier *toto.txt* (*inode 100*)

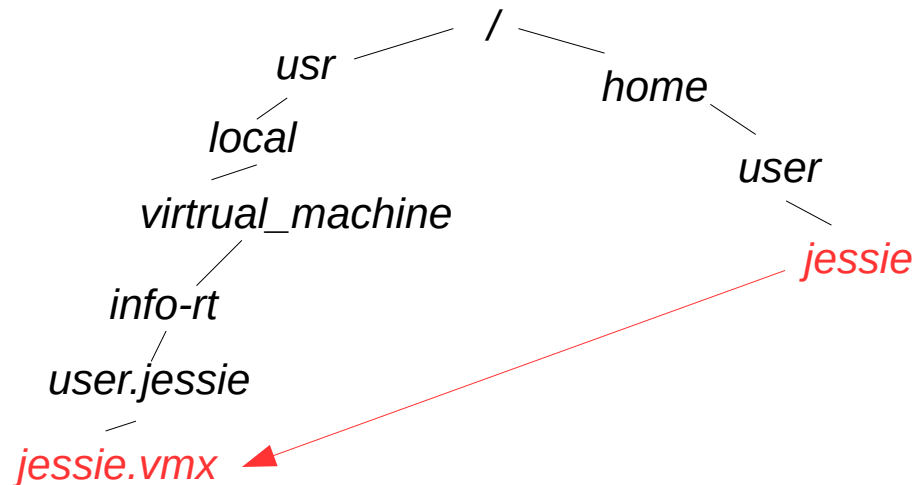


Contenu de l'inode

Propriétaire du fichier
Groupe du propriétaire du fichier
Type du fichier
Mode d'accès au fichier
Date du dernier accès au fichier (<i>lecture</i>)
Date de la dernière modification du fichier (<i>écriture</i>)
Date de la modification du noeud d'index (<i>droits</i>)
Nombre de liens du fichier
Taille du fichier
Table des adresses des données (numéros de blocs)

Les liens

- Pour faciliter, l'administration il peut être intéressant de créer des raccourcis
 - un nom court par rapport à un chemin dans l'arborescence



=> il existe plusieurs noms pour désigner une même ressource

ls /usr/local/virtual_machine/info-rt/user.jessie/jessie.vmx

<=>

ls ~user/jessie

Les liens physiques

- 2 types de liens
 - Lien physique
 - Lien symbolique
- **Le lien physique :**
 - plusieurs noms de fichiers sont associés à la même inode
 - ne traverse pas les partitions, pas sur les répertoires
- Commande de création : **ln** *chemin_fichier_existant nouveau_nom*
- Un fichier n'est supprimé que lorsque tous les liens physiques donnant accès à son contenu sont supprimés

Liens physiques

- Commande de listage : **ls -li**

inode Total 1 nombre de liens

```

456 -rwxr-xr-x 1 util1 512 Jul 8 15:08 unFichier
ln unFichier unAutreFichier
ls -li
456 -rwxr-xr-x 2 util1 512 Jul 8 15:08 unFichier
456 -rwxr-xr-x 2 util1 512 Jul 8 15:08 unAutreFichier
  
```

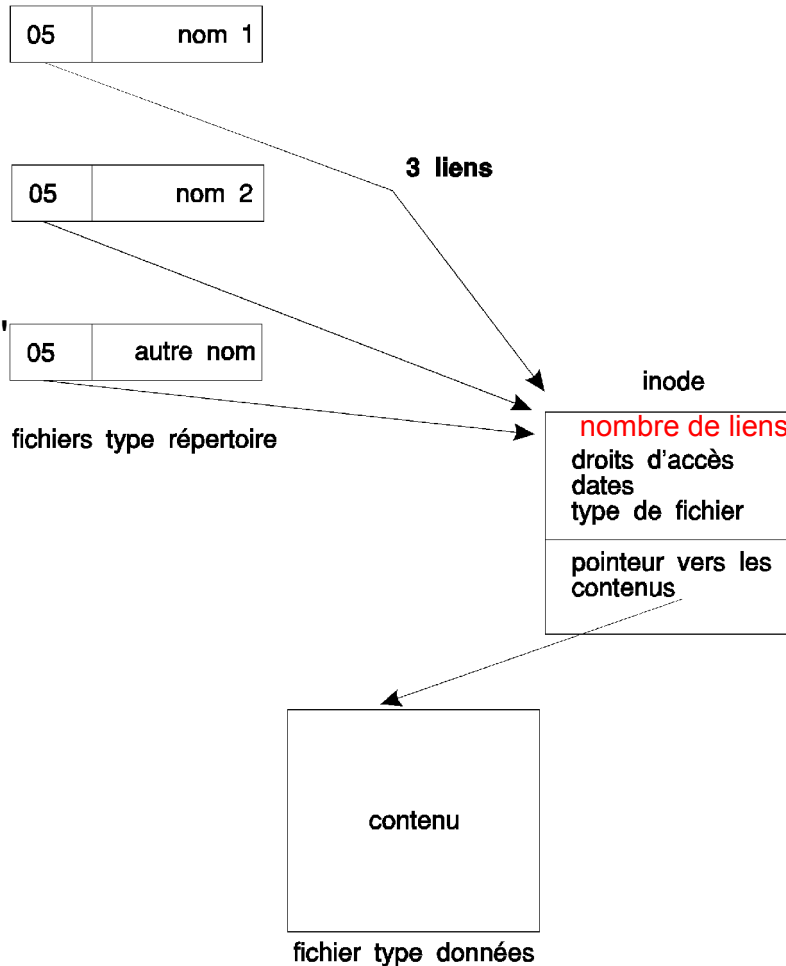
Diagram annotations: Red circles around the inode numbers (456) and the link counts (1 and 2). Red arrows point from the text 'nombre de liens' to the link counts 1 and 2.

- Répertoires
 - Un répertoire a un nombre de liens au moins égal à 2
 - désignation : son nom + '.'
 - Le nombre de liens est augmenté de 1 par répertoire fils contenu
 - désignation : '..'

Liens physiques et inodes

Création des fichiers :
touch "nom 1"
ln "nom 1" "nom 2"
ln "nom 2" "autre nom"

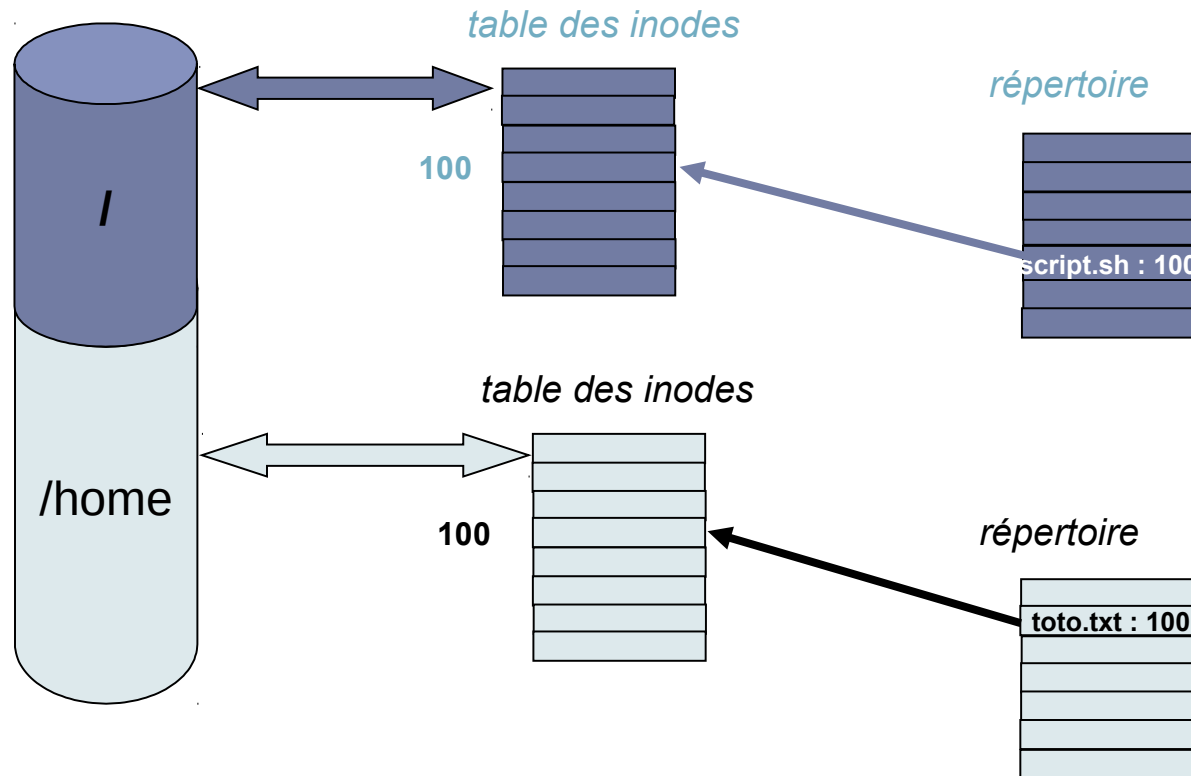
→ nb liens = 3
(pour les 3 fichiers)



influence sur le nb de liens

- ln
 - nb liens ++
- mv
 - pas d'impact
- rm
 - nb liens --
- cp
 - pas d'impact
 - inodes ≠

Inodes relatifs à un système de fichiers



Les liens symboliques

- Limites du lien physique : reste au sein d'une même partition
 - *puisque un numéro d'inode n'a de sens que pour un système de fichier unique*
- Alors que le lien physique utilise le numéro d'inode le lien symbolique utilise le chemin d'accès et lève ainsi les limitations du lien physique :
 - permet de créer un lien traversant les partitions
 - permet de créer un lien sur un répertoire
- Commande de création : **ln -s** *chemin nomLienSymbolique*
nomLienSymbolique est un nouveau fichier *texte* qui contient un chemin
- Commande de listage : **ls -l**

enregistrement du chemin

```
lrwxrwxrwx 1 util1 10 Dec 9 19:12 liensymb -> ../rep/fic
```


Gestion des liens symboliques

- Pour créer un lien symbolique il faut de préférence utiliser un adressage absolu pour spécifier la cible (*le chemin reste valide après déplacement*)
 - Pour pouvoir utiliser un adressage relatif
 - il faut se placer au préalable dans le répertoire où sera créé le lien
 - si par la suite ce lien est déplacé, il n'aura plus de sens (chemin exprimé à partir du répertoire d'origine)
- => Il peut exister dans le système des liens symboliques qui ne pointent plus sur une cible existante
- ➔ lors de la suppression d'un fichier il faudrait supprimer également les liens symboliques dont il est la cible (i.e. parcourir l'arborescence et contrôler les chemins...)

La commande find

- Rechercher des fichiers dans une **sous-arborescence** selon plusieurs critères
 - Parcourt récursivement un ou plusieurs répertoires afin d'y trouver des fichiers correspondant à un ensemble de critères

find [options] [répertoires] [critère]

par défaut : recherche à partir du répertoire courant

- Quelques critères :
 - -name nom du fichier recherché
 - -inum numéro d'inode
 - -type type du fichier
 - ...
- Exemple :
 - find / -user util1 : recherche les fichiers appartenant à l'utilisateur util1

Exécutions multiples

- Exécution d'une commande sur les fichiers correspondants aux critères
find [options] [répertoires] [critère] **[action]**
- Quelques actions :
 - -print affiche nom des fichiers trouvés (avant exec)
 - -delete détruit les fichiers trouvés
 - -exec cmd {} \; exécute la commande cmd sur les fichiers trouvés
 - -ok cmd demande confirmation avant l'exécution
 - ...
- Exemple :
 - **find ~ -name "*.tmp" -exec rm {} \;**
 - supprime les fichiers d'extension .tmp de l'arborescence de l'utilisateur connecté