

CHAPITRE 5 : Gestion des fichiers

Mohammed SABER

Département Électronique, Informatique et Télécommunications
École Nationale des Sciences Appliquées "ENSA"
Université Mohammed Premier OUJDA

Année Universitaire : 2018-2019

Plan de chapitre

1 Introduction

2 Arborescence Linux des fichiers

3 Les inodes

4 Les chemins des fichiers

5 Les liens des fichiers

6 Les fichiers

Plan de chapitre

1 Introduction

2 Arborescence Linux des fichiers

3 Les inodes

4 Les chemins des fichiers

5 Les liens des fichiers

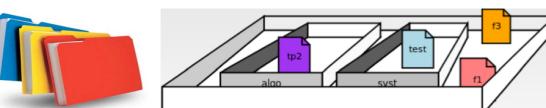
6 Les fichiers

Introduction

- On entend par fichier sous linux, la structure contenant des données utilisateur.
- Les fichiers standard sont constitués d'une suite de caractères ou flux d'octets dont le format n'est pas imposé par le système mais par les applications.



- Les répertoires sont des fichiers particuliers pouvant contenir plusieurs autres fichiers, répertoires ou non. Cela permet d'organiser les fichiers de façon hiérarchique et arborescente.



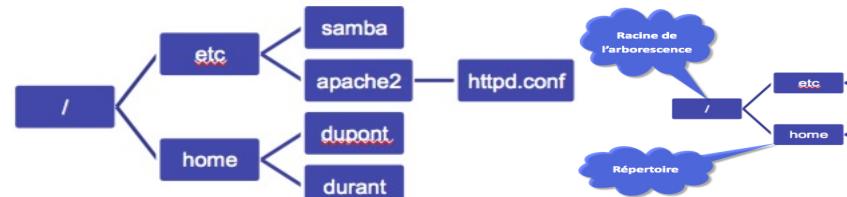
- 1 Introduction
- 2 Arborescence Linux des fichiers
- 3 Les inodes
- 4 Les chemins des fichiers
- 5 Les liens des fichiers
- 6 Les fichiers

Qu'est-ce qu'un système de fichier ?

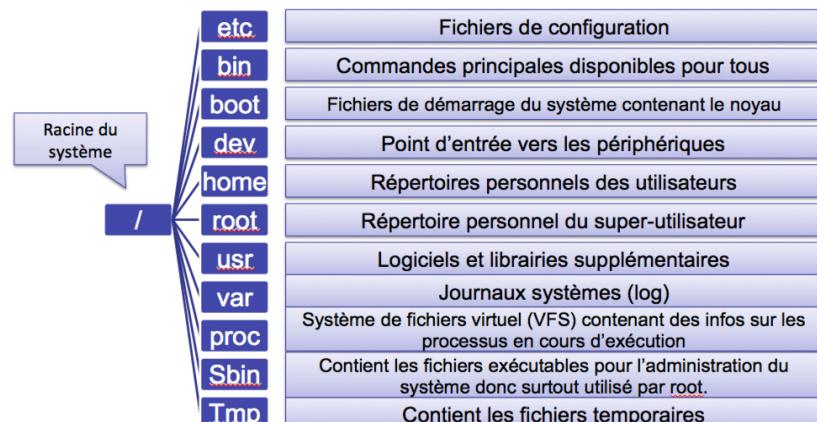
- Organisation physique des données sur un support (Sur un disque dur, une clé USB, un DVD, ...).
- Exemple : FAT32, NTFS, EXT3, EXT4,

Qu'est-ce qu'une arborescence ?

- Organisation logique des fichiers sur un ou plusieurs systèmes de fichiers.
- Il s'agit d'une structure de données hiérarchique de type arbre.



- Les fichiers du système Linux sont organisés en arbre.
- Linux divise les fichiers en plusieurs catégories.



- Les plus importantes du point de vue de l'organisation et du fonctionnement du système sont les répertoires qui structurent l'arborescence, les fichiers ordinaires qui contiennent les programmes et les données, et les fichiers spéciaux de type périphérique.
- Ces derniers permettent aux utilisateurs, y compris l'administrateur, d'avoir une vision uniforme.
- **Tout est fichier**, même les périphériques, qui ont un nom dans l'arborescence et qui sont donc désignés par un chemin d'accès comme le sont les fichiers ordinaires.
- L'administrateur doit mémoriser les noms et définir le contenu des principaux répertoires, ceux qui sont le plus fréquemment utilisés dans l'administration quotidienne d'un système Linux.
- Les fichiers sont dans un seul arbre. Les utilisateurs n'ont pas besoin de mentionner le nom du disque où ils résident, ce qu'ils ignorent souvent.
- Cette vision d'un arbre unique s'étend également aux répertoires partagés à travers le réseau.
- C'est l'administrateur qui gère les arbres de chaque disque et les attache les uns aux autres. On parle de **montage**.

- L'invite de commandes de la console contient les informations suivantes :



- Les symboles suivants ont une signification particulière :

• « . » : Le point désigne le répertoire courant (Exemple : Exécuter un script depuis le répertoire courant).

```
root@Nom_machine :/home/saber# ./script.sh
```

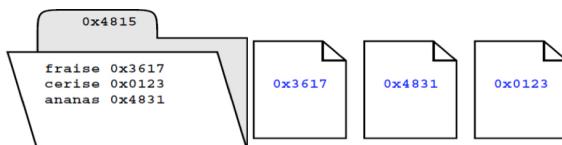
• « .. » : Les 2 points désignent le répertoire parent (Exemple : Se déplacer dans le répertoire parent).

```
root@Nom_machine :/home/saber/Docs# cd ../../
```

• « ~ » : Désigne le répertoire home de l'utilisateur courant

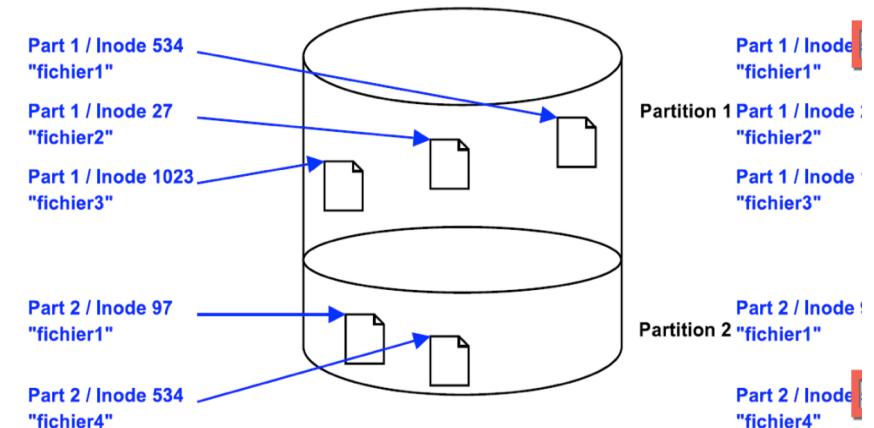
```
root@Nom_machine :/home/saber/Docs/Cours/AminSys# cd ~
```

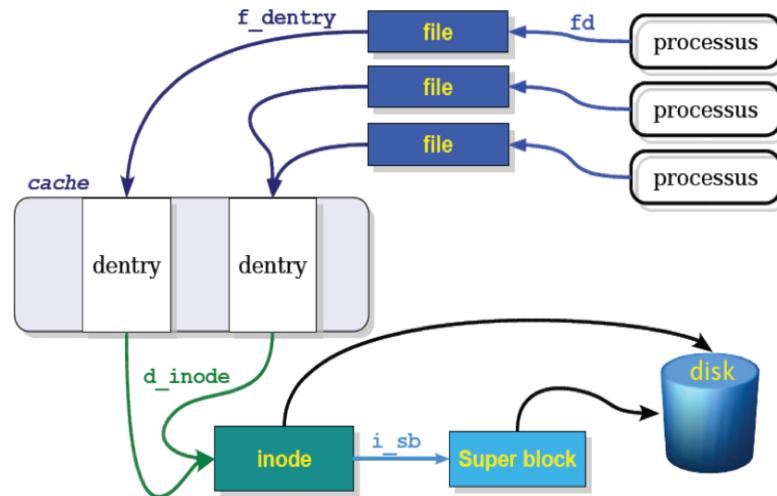
- Inode = Index Node.
- Les objets sont manipulables sur le disque dur via l'intermédiaire d'une structure de données appelée « inode ».
- Un objet est identifié sur une partition de disque dur par son numéro d'inode et non pas par son nom.
- L'inode n'est unique qu'au sein d'une même partition.
- L'inode d'un objet ne stocke pas le nom de l'objet !
- Le nom d'un objet n'est pas stocké dans l'inode. Le nom d'un objet est stocké dans le répertoire.
- Un répertoire = liste de couples (nom d'objet, numéro d'inode).



- 1 Introduction
- 2 Arborescence Linux des fichiers
- 3 Les inodes
- 4 Les chemins des fichiers
- 5 Les liens des fichiers
- 6 Les fichiers

- Le système manipule en fait les objets via un couple (partition, inode).

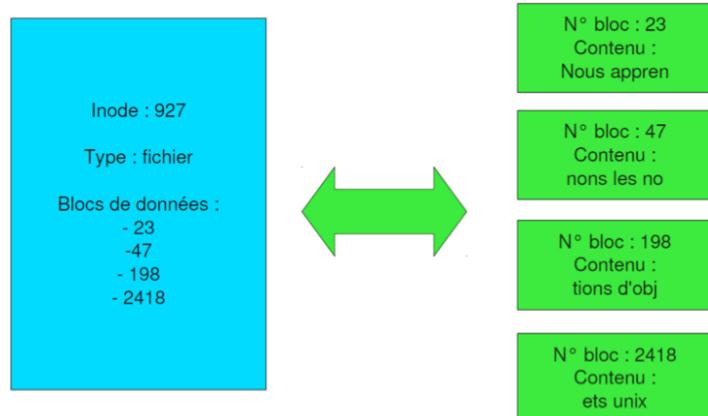




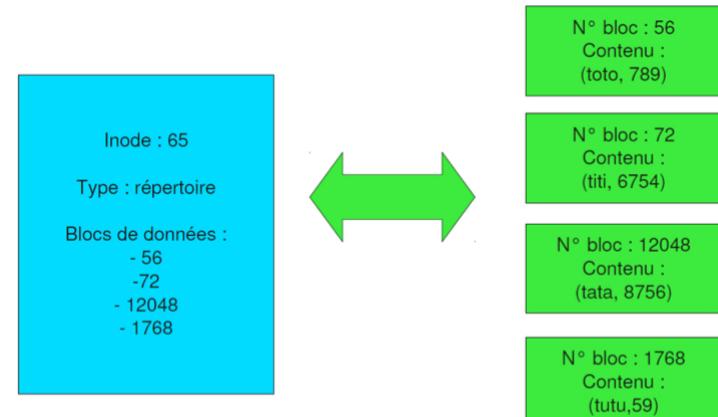
- Un « inode » est une structure de données concernant un fichier. Contient des informations sur :
 - Un inode a un numéro unique ;
 - Un inode indique le type de l'objet ;
 - Un inode possède la liste des blocs de données de l'objet ;
 - Les droits, le propriétaire et le groupe ;
 - Le périphérique qui le contient ;
 - Des données relatives au système de fichiers et à l'emplacement du fichier sur le support de stockage ;
- A chaque fichier, correspond un « **inode** » ;
- Pour connaître l'inode d'un fichier, la commande « `ls` » avec l'option «`-i` ».

```
root@Nom_machine :# ls -i
8246 drwxr-xr-x 2 root root 4096 2008-08-25 15 :08 essais
467165 drwxr-xr-x 3 root root 4096 2008-06-02 14 :20 software
475969 drwxr-xr-x 3 root root 4096 2008-07-29 15 :54 vmware-tools
```

- Un fichier correspond à un inode de type fichier.



- Sous Linux, un répertoire est aussi un fichier, ce fichier particulier contient une liste d'associations : (nom de fichier , inode).



- 1 Introduction
- 2 Arborescence Linux des fichiers
- 3 Les inodes
- 4 Les chemins des fichiers
- 5 Les liens des fichiers
- 6 Les fichiers

Chemin absolu	Chemin relatif
<code>root@Nom_machine :/home/saber# cd /home/saber/Docs</code>	<code>root@Nom_machine :/home/saber# cd Docs</code>
<code>root@Nom_machine :/home/saber# cd /home/saber/Docs</code>	<code>root@Nom_machine :/home/saber# cd Docs</code>
<code>root@Nom_machine :/home/saber# cd /home/saber/Docs#</code>	<code>root@Nom_machine :/home/saber/Docs#</code>
<code>root@Nom_machine :/etc# cd /home/saber/Docs</code>	<code>root@Nom_machine :# cd Docs</code>
<code>root@Nom_machine :/home/saber# cd /home/saber/Docs#</code>	<code>root@Nom_machine :/home/saber/Docs#</code>

Il existe 2 méthodes pour spécifier un chemin dans le système de fichiers.

Chemin absolu

- C'est le chemin qui permet d'accéder à un fichier et qui commence par la racine de l'arbre.
- Un chemin absolu doit toujours commencer par «/».

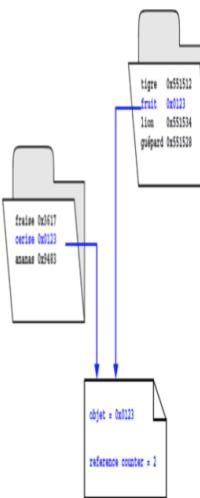
Chemin relatif

- Un chemin relatif est «relatif» à la position de référence (le répertoire courant) ;
- Le répertoire courant est noté «.» ;
- Le répertoire parent du répertoire courant est noté «..».

- 1 Introduction
- 2 Arborescence Linux des fichiers
- 3 Les inodes
- 4 Les chemins des fichiers
- 5 Les liens des fichiers
- 6 Les fichiers

- Un nom est appelé un lien sur l'objet ;
- Sur Linux, chaque objet peuvent être associés plusieurs noms ;
- Dans l'inode d'un objet, il y a un compteur de liens :
 - Compteur incrémenté lors de la création d'un nouveau lien ;
 - Compteur décrémenté lors de la suppression d'un lien ;
 - L'objet est détruit lorsque le dernier lien sur l'objet est supprimé ;
 - « . » et « .. » sont des liens.
- On voit les valeurs des compteurs de liens via la commande «ls -l» :

```
root@Nom_machine :# ls -l
-rw-r-r- 1 saber ensao 39 Oct 26 2013 ensao.txt
-rw-r-r- 1 saber ensao 35 Jul 3 17:38 contre-fusion.txt
drwxr-xr-x 4 saber ensao 512 Jul 4 15:48 contre_fusion
drwxr-xr-x 3 saber ensao 512 Jul 5 00:36 saber
```



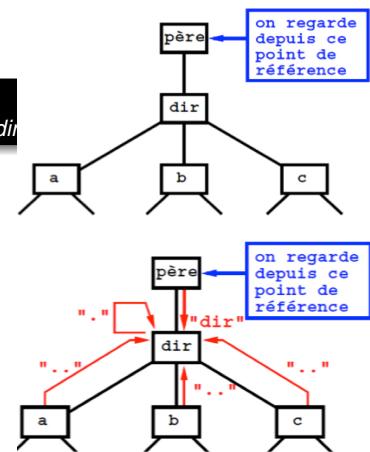
- Soit l'arborescence :

```
root@Nom_machine :# ls -l
drwxr-xr-x 5 saber ensao 512 Jul 5 00:29 dir
```

- Pourquoi a-t-on l'indication de 5 liens sur « dir » ?

- Ces 5 noms sont :

- lien «/chemin/vers/dir»;
- lien «/chemin/vers/dir/.»;
- lien «/chemin/vers/dir/a/..»;
- lien «/chemin/vers/dir/b/..»;
- lien «/chemin/vers/dir/c/..».

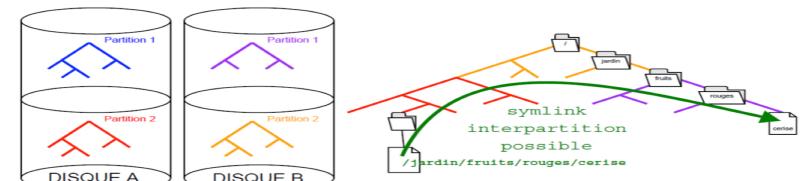


- Preuve via l'utilisation de l'option «-i» de « ls » qui affiche les numéros d'inodes :

```
root@Nom_machine :# ls -ldi dir dir/. dir/a/.. dir/b/..
dir/c/..
550907 drwxr-xr-x 5 saber ensao 512 Oct 5 00:29 dir
550907 drwxr-xr-x 5 saber ensao 512 Oct 5 00:29 dir/.
550907 drwxr-xr-x 5 saber ensao 512 Oct 5 00:29 dir/a/..
550907 drwxr-xr-x 5 saber ensao 512 Oct 5 00:29 dir/b/..
550907 drwxr-xr-x 5 saber ensao 512 Oct 5 00:29 dir/c/..
```

- Sous Linux, il existe deux sortes de liens :
 - Liens **symboliques** (aussi appelés symlinks) ;
 - Liens en dur (aussi appelés liens matériels, ou liens physiques ou même, dans certains messages d'erreur liens directs ; en anglais hard links).

- Un lien symbolique est non limité à une partition d'un disque dur parce qu'utilisant le chemin d'un objet et non pas son numéro d'inode ;



- Le lien symbolique est une référence (chemin) vers un fichier cible ;
 - Lorsque le fichier cible est effacé, le lien est rompu ;
 - Lorsque le lien est effacé, le fichier cible n'est pas effacé ;
- La commande «ln» avec l'option «-s» est utilisée pour créer un lien symbolique ;

```
root@Nom_machine :# ln -s /proc/sys/net/ipv4/ip_forward ip
root@Nom_machine :# ls -l
lrwxrwxrwx 1 saber ensao 29 2015-09-25 14:23 ip ->
/proc/sys/net/ipv4/ip_forward
```

- Un lien physique est associé à un emplacement sur le support de stockage ;
 - Deux liens associés au même «inode» ;
 - Similaire à la notion de «pointeur» du langage C ;
 - Deux liens physiques sont considérés comme deux fichiers indépendants ;
 - Le lien physique est vu comme un fichier régulier ;
- Créer un lien physique avec la commande «ln» :

```
root@Nom_machine :# ln
/home/saber/Documents/contre-fusion-2016-2017.doc
ensao1617
root@Nom_machine :# ls -il
470930 -rw-r--r-- 2 saber ensao 84091 2015-05-25 14:48 ensao1617
470930 -rw-r--r-- 2 saber ensao 84091 2015-05-25 14:48
ontre-fusion-2016-2017.doc
```

- Exemple : même caractéristiques (type, inode, date, taille, ...) pour les deux fichiers.

- 1 Introduction
- 2 Arborescence Linux des fichiers
- 3 Les inodes
- 4 Les chemins des fichiers
- 5 Les liens des fichiers
- 6 Les fichiers

- La différence entre les deux types de lien apparaît de façon immédiatement visible dans l'affichage résultant de la commande ls utilisée avec les options appropriées,
- On peut créer les deux sortes de liens à l'aide de la commande ln. Avec l'option -s un lien symbolique sera créé, sans cette option on obtiendra un lien en dur.

```
1606777 -rw-r--r-- 1 saber ensao 3407 Sep 27 12:57 ensao.txt
root@Nom_machine :# ln -s ensao.txt ensa1516
root@Nom_machine :# ln ensao.txt ensa.txt
root@Nom_machine :# ls -il
1606777 -rw-r--r-- 2 saber ensao 3407 Sep 27 12:57 ensao.txt
1606777 -rw-r--r-- 2 saber ensao 3407 Sep 27 12:57 ensa.txt
1610496 lrwxrwxrwx 1 saber ensao 14 Sep 28 14:52 ensao1617 -> ensao.txt
```

- Le fichier d'origine et le lien en dur :
 - Le même numéro d'inode (alors que le lien symbolique a un numéro d'inode qui lui est propre) ;
 - Le même type de fichier, ici «fichier régulier» indiqué par le tiret en tête des permissions, alors que le lien symbolique a un type différent (l pour : «lien symbolique») ;

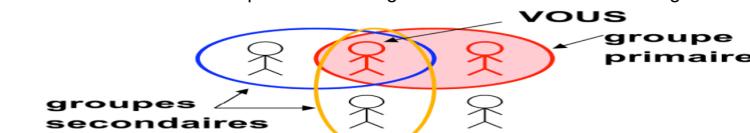
Il existe en fait sept types sous Linux, repérés par une lettre différente en début de ligne à l'affichage de la commande ls -l :

Type des fichiers

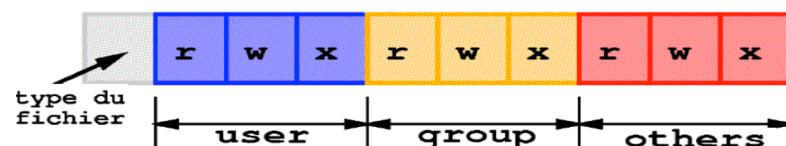
Fichier	Symbolé	Création	Destruction
Fichier standard	-	vi, touch, nano, ...	rm
Répertoire (directory)	d	mkdir	rmdir, rm -r
Périphérique bloc	b	mknod	rm
Périphérique caractère	c	mknod	rm
Lien symbolique ou logique (soft link)	l	ln	rm
Tube nommé (named pipe)	p	mknod	rm
Socket locale	s	socket	rm

- Répertoire** : un fichier qui contient d'autres fichiers.
- Les périphériques de caractère** : les périphériques où l'unité d'échange est le caractère, typiquement les liaisons séries.
- Les périphériques de bloc** : les périphériques qui échangent des blocs, typiquement les disques (qui fonctionnent aussi en mode caractère).
- Socket locale** : Le terme « socket » désigne un mécanisme d'échange d'information entre tâches locales ou distantes (en réseau) implémenté dans le noyau Linux.
- Tube nommé** : Un tube nommé est un mécanisme d'échange d'information entre tâches qui s'exécutent localement (|).
- Un lien symbolique** : un lien symbolique est un fichier spécial qui définit un chaînage sur un autre fichier.

- Linux est un système multi-utilisateurs.
 - Plusieurs utilisateurs se partagent l'espace disque.
 - Les fichiers et répertoires d'un utilisateur ne doivent pas être accessibles par les autres.
 - Les fichiers de configuration du système doivent être protégés.
 - Les droits sont responsables d'un grand nombre d'erreurs de configuration.



- Les droits d'accès sont stockés dans la structure inode du fichier.
 - Les droits sont affichés par la commande «ls -l».
 - Ils sont représentés par les 10 premiers caractères de la commande «ls -l».



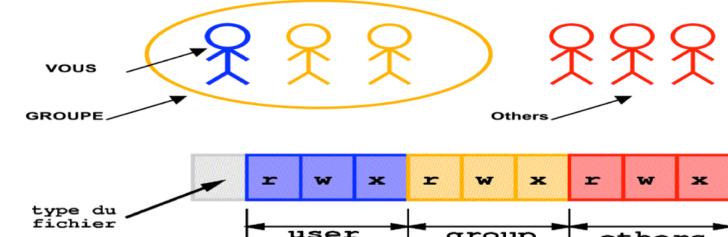
- La commande «ls -l» permet d'afficher les attributs qui s'appliquent.

```
root@Nom_machine :# ls -l
-rw-r-r- 2 saber ensao 3407 Sep 27 12:57 ensao.txt
-rw-r-r- 2 saber ensao 3407 Sep 27 12:57 ensa.txt
-rwxrwxrwx 1 saber ensao 255 Sep 15 14:52 script.sh
```

- Signification des différents champs.



- Cas le plus courant : l'utilisateur appartient au groupe propriétaire ;



- Nécessité de spécifier des droits pour chaque fichier/répertoire ;
 - Ces droits s'appliquent pour 3 groupes d'utilisateurs ;
 - Droits du propriétaire (**u - user**) ;
 - Droits des membres du groupe (**g - group**). (Tous les utilisateurs membres du groupe) ;
 - Droits des autres utilisateurs (**o - others**). Désigne tous les utilisateurs non membres des 2 précédents ;
 - Il existe 3 droits d'accès (permissions) associés à chaque objet ;
 - Droits en lecture (**r - read**) ;
 - Droits en écriture (**w - write**) ;
 - Droits en exécution (**x - execute access**) ;

Ces droits n'ont pas la même signification pour un fichier que pour un répertoire

- Pour un fichier :
 - **r** : Lecture (afficher) ;
 - **w** : Ecriture (modification) ;
 - **x** : Exécution (exécution d'un script) ;
 - **-** : aucun droit ;
- Pour un répertoire :
 - **r** : Lire le contenu, lister les fichiers (avec `ls` par exemple) ;
 - **w** : Modifier le contenu, créer et supprimer des fichiers (avec les commandes «`cp`», «`mv`», «`rm`») ;
 - **x** : Permet d'accéder aux fichiers du répertoire. Mais aussi de naviguer dans les sous-répertoires (avec «`cd`») ;
 - **-** : aucun droit ;
- En général, le droit **w** est souvent associé au droit **x** ;

La commande «`chmod`» permet de modifier les droits :

Mode symbolique

- Basé sur des symboles (ugoa) et des opérateurs (+,-,=) ;
- **u** (user), **g** (group), **o** (others), **a** (all users) ;
- **+** (Ajouter le droit), **-** (Retirer le droit), **=** (Ajouter le droit et retirer tous les autres) ;
- Exemple (Ajoute le droit d'exécution au propriétaire) : `chmod u+x rapport.txt` ;

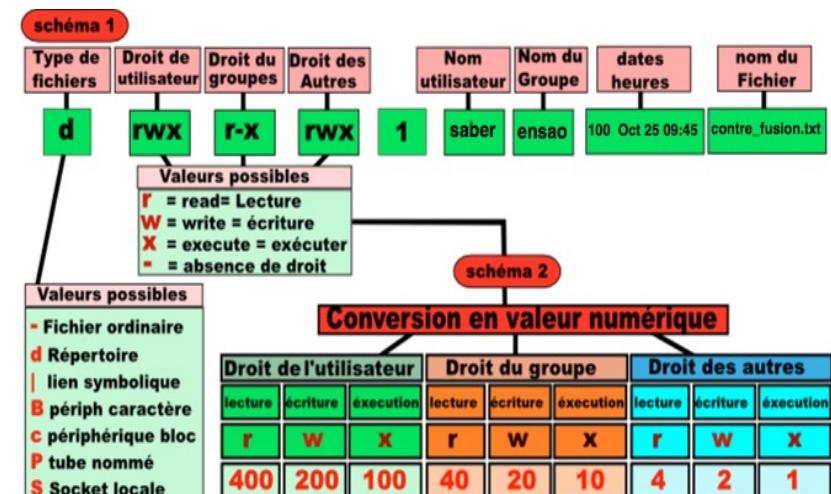
Mode octal

- Basé sur des nombres de 0 à 7 ;
- A chaque bit de la traduction binaire correspond un droit
- Exemple (**rw- rw- r-**) : `chmod 664 rapport.txt` ;

- Les droits sont représentés par un nombre octal (Base 8), de 1 à 7 ;
- La représentation binaire (base 2) donne le détail des droits ;

Droits	Valeur en octal
---	0
--x	1
-w-	2
-wx	3
r--	4
r-x	5
r-w	6
rwx	7

- Ce mode permet de modifier tous les droits en même temps ;
 - A utiliser avec précaution ;
 - Très efficace pour s'assurer que tous les fichiers ont les mêmes droits ;
 - Utilisé pour sécuriser les accès des utilisateurs aux fichiers ;



- Utilisé pour définir les droits par défaut : droits appliqués pour un nouveau fichier lors de sa création ;
- Les bits du masque à 1 empêchent le fichier d'obtenir le droit correspondant ;

Droits	Valeur en octal
-- --	7
-- x	6
-w -	5
-wx	4
r --	3
r-x	2
r w-	1
rwx	0

- Exemple avec un masque de protection de **027** ;

0 2 7
 ↓ ↓ ↓
 000 010 111

rwX rwX rwX Permissions maximum

rwx r-x --- Permissions effectives après application du masque

- La commande «umask» permet de modifier le masque ;
 - Les fichiers et répertoires nouvellement créés seront alors protégés ;
 - La valeur par défaut du masque est **022** ;

- chown [Rh] user objet
 - Change le propriétaire des objets ;
 - Option «R» : changement réursif dans une arborescence ;
 - Option «h» : en cas de lien symbolique, change le propriétaire du lien et non les objets pointés par le lien ;
 - Commande réservée à **root** (sinon problème de sécurité) ;

```
root@Nom_machine :# ls -l ensao.txt
-rw-r-r- 2 saber ensao 3407 Sep 27 12:57 ensao.txt
root@Nom_machine :# chown ensao ensao.txt
-rw-r-r- 2 ensao ensao 3407 Sep 27 12:57 ensao.txt
```

- chgrp [Rh] groupe objet
 - Faire appartenir un fichier à un groupe ;
 - Change le groupe primaire des objets ;
 - Fonctionnement identique à **chown** (il faut être **root** ou appartenir au groupe) ;

```
root@Nom_machine :# ls -l ensao.txt
-rw-r-r- 2 ensao ensao 3407 Sep 27 12:57 ensao.txt
root@Nom_machine :# chgrp ump ensao.txt
-rw-r-r- 2 ensao ump 3407 Sep 27 12:57 ensao.txt
```

Manipulation des fichiers et des répertoires

- Répertoire** : mkdir, rmdir, mv, cp, pwd, cd, find.
- Fichier** : touch, rm, mv, cp, locate, gedit, find, nano, vi, cat, , sort, , more, less, head, tail, cmp, grep, wc, sed.
- Pour le détail de chaque commande voir le manuel par **man commande**.

QUESTIONS ?