Les droits d'accès aux fichiers

- La combinaison des droits d'accès à une ressource constitue son mode d'accès
- Chaque fichier possède un mode d'accès
- Le propriétaire du fichier est le seul autorisé à modifier ce mode d'accès (hormis root)
- Le mode d'accès est toujours spécifié vis-à-vis du contenu du fichier (répertoire, normal, périphérique)

Les droits d’accès aux fichiers

Les droits d'accès sont fixés en fonction du type d'utilisateur / propriétaire

Pour un fichier, on distingue :

- **3 types d'utilisateurs :**
  - **le propriétaire (user U)**
  - **les personnes du groupe du propriétaire (group G)**
  - **les autres (other O)**
- **3 types de permissions principaux :**
  - **lecture (r)**
  - **écriture (w)**
  - **exécution (x)**
- **Changer le mode : commande chmod**
- **Changer le propriétaire : commande chown**
- **Changer le groupe : commande chgrp**

Les droits d'accès aux fichiers

- **Droit r :**
  - **répertoire : affichage du contenu**
  - **fichier : affichage du contenu**
- **Droit w :**
  - **répertoire : modification du contenu du répertoire**
  - **fichier : modification du contenu**
- **Droit x :**
  - **répertoire : droit de traversée**
  - **fichier : droit d'exécution**

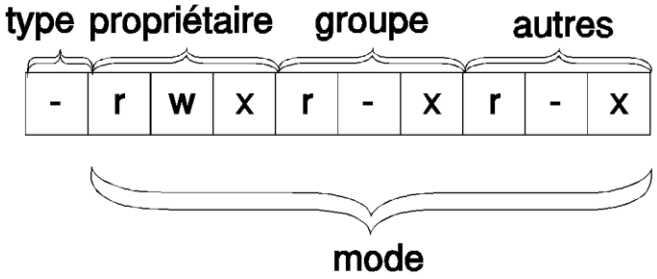
Protéger un fichier :

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

Visualiser le mode d'accès

- **La commande ls –lg**

```
$ ls –lg
Total 1
-rwxr-xr-x 1 util1 users  512  Jul 8  15:08  unFichier
```



- **Les types**

-	fichier normal
d	répertoire
c, b	spécial

Le droit « s » : droit d'endossement

- **Pour un fichier exécutable, le droit s peut remplacer le droit x au niveau du propriétaire ou de groupe :**

  un utilisateur « autre » hérite alors des droits (droits effectifs) du propriétaire (EUID) ou du groupe (GUID) pour l'exécution de cet exécutable.

  Par défaut, un programme accède aux fichiers avec les droits de l'utilisateur l'ayant lancé.

- **les commandes setuid et setgid permettent de spécifier les droits effectifs du processus courant**

Le droit « s »

- Ce droit est utilisé par exemple par la commande passwd (/usr/bin) pour permettre à un utilisateur de changer son mot de passe (fichier /etc/passwd)

- Le droit S (majuscule) correspond aux fichiers non exécutables

- Pour un répertoire, un droit s pour le groupe, signifie que les fichiers du répertoire seront créés comme appartenant au groupe du propriétaire du répertoire

Fichiers persistants

- **Le droit « t » : sticky bit**
- **Pour un fichier exécutable :**
  - **il reste en mémoire, le chargement est très rapide.**
  - **Ne pas abuser de ce droit.**
- **Pour un répertoire :**
  - **Seuls les propriétaires des fichiers du répertoire peuvent les détruire.**
  - **par défaut, c'est le propriétaire du répertoire qui a ce droit**

Changer le mode d'accès : notation symbolique

Utilisation de la commande « chmod »

Classes	Opérations	Permissions
u : utilisateur g : groupe o : autres a : tous	= : affectation d'une permission - : suppression de droits + : ajout de droits	r : lecture w : écriture x : exécution s : endossement t : sticky bit

Exemple :

```
chmod a+r fichier
chmod +x, g=  fichier
```

Applications des droits avec les nombres décimaux

- On peut appliquer des droits sur les fichiers ou répertoires des trois types de classes (u,g et o) à l'aide de la commande chmod suivi de 3 ou 4 nombres décimaux, ceux-ci correspondants à une valeur binaire. In fine, c'est la valeur binaire qui détermine quel droit va être appliqué.

On utilise la commande chmod de cette façon :
chmod nnnn fichier ou bien chmod nnn fichier

NB: On utilise plus couramment 3 chiffres, car le premier des quatre chiffres lorsqu'il est utilisé correspond aux droits spécifiques « suid et sticky bit ».

- A chaque permission est associée une valeur décimale :

valeur	permission
4	r
2	w
1	x
0	aucune

- Exemple : droit rw- : 6 (valeur décimale)
- Les permissions doivent être précisées pour toutes les classes d'utilisateur

  Ex : propriétaire rwx, groupe rw, autres r => chmod 0764 toto
- Le premier octet représente les droits spéciaux
  - 4 : setuid, 2 :setgid, 1 : sticky bit

Applications des droits avec les nombres décimaux

Ce tableau ci-dessous représente les nombres décimaux que l'on utilise lors d'une attribution de droits avec la commande chmod. Cette représentation s'applique aux trois derniers chiffres (quand quatre sont utilisés) ou bien les trois premiers chiffres lors que le choix est fait d'en utiliser trois seulement.
(Exemple chmod 0666 ou bien chmod 666). A un nombre décimale correspond un nombre binaire auquel in fine correspond une représentation des droits qui seront appliqué avec le nombre décimal choisi.

nombre décimal	nombre binaire	signification
0	0 0 0	- - -
1	0 0 1	- - x
2	0 1 0	- w -
3	0 1 1	- w x
4	1 0 0	r - -
5	1 0 1	r - x
6	1 1 0	r w -
7	1 1 1	r w x

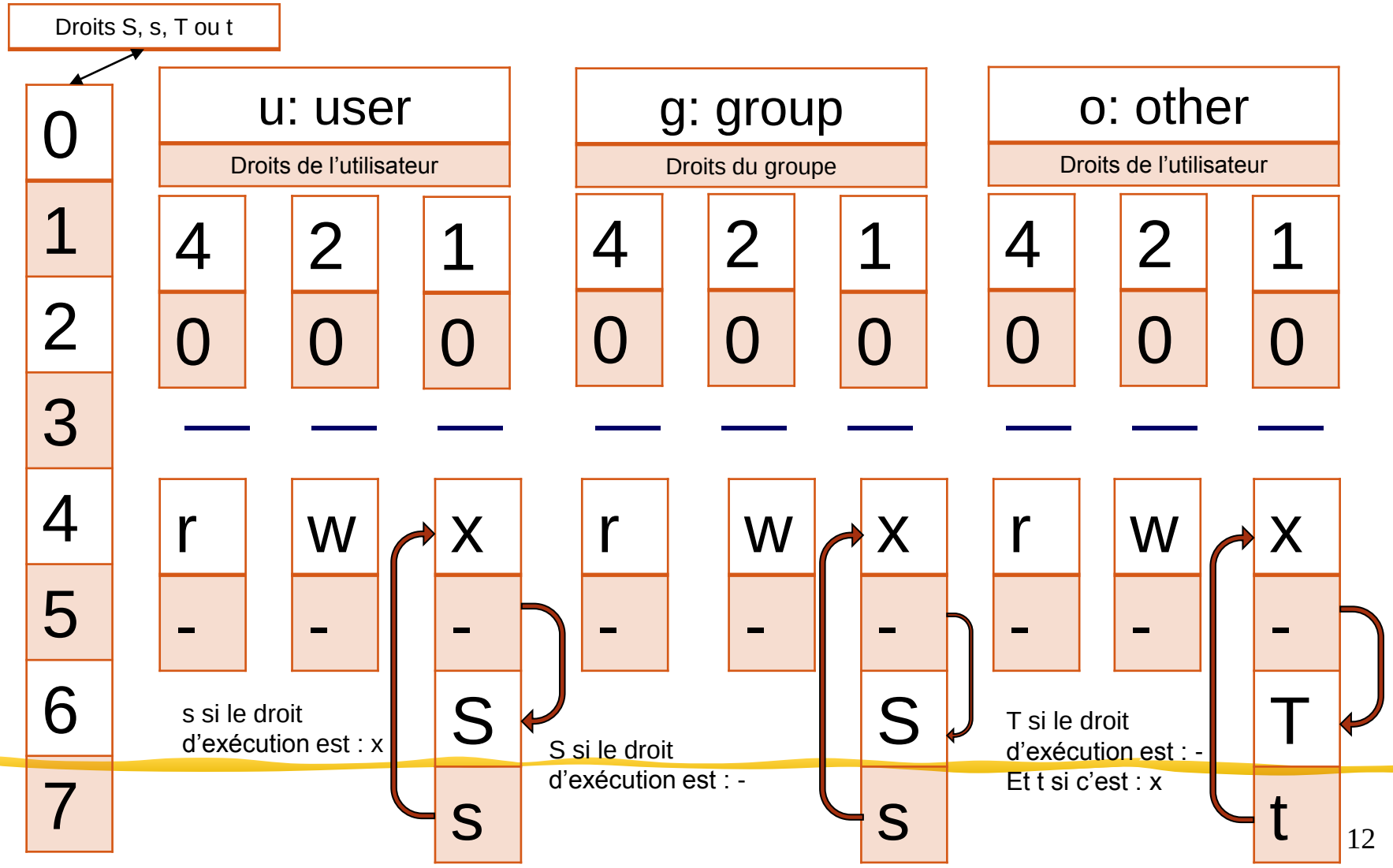
Applications des droits avec les nombres décimaux

Le tableau ci-dessous indique la valeur à choisir du premier des quatre chiffres lorsqu'on souhaite appliquer les droits spécifiques « suid ou sticky bit » avec la commande chmod.

nb. décimal	nb. binaire	x (1ère série)	x (2ème série)	x (3ème série)
0	0 0 0	inchangé	inchangé	inchangé
1	0 0 1	inchangé	inchangé	t
2	0 1 0	inchangé	s	inchangé
3	0 1 1	inchangé	s	t
4	1 0 0	s	inchangé	inchangé
5	1 0 1	s	inchangé	t
6	1 1 0	s	s	inchangé
7	1 1 1	s	s	t

Toutes les valeurs possibles des droits décimaux sur Linux

Exemple d'application des droits: chmod 0000 fichier* ou *chmod 7777 fichier.
Chaque chiffre décimal peut prendre les valeurs de 0 à 7.



Toutes les valeurs possibles des droits décimaux sur Linux

On peut appliquer jusqu'à **4096 droits sur linux avec l'utilisation de 4 chiffres décimaux** dans le chmod ou bien **512 droits avec 3 chiffres décimaux**.
En fait il s'agit de l'utilisation de 12 bits ou bien 9 bits.

Droits spéciaux (s,S,t,T)			
Puissances binaires			Valeur décimale
2^2	2^1	2^0	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Droits de « l'user » dits « u » ou utilisateur			
Puissances binaires			Valeur décimale
2^2	2^1	2^0	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Droits de « group » dit « g » ou groupe			
Puissances binaires			Valeur décimale
2^2	2^1	2^0	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Droits de « other » dit « o » ou autres			
Puissances binaires			Valeur décimale
2^2	2^1	2^0	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

3 bits = 2^3 = 8 choix possibles | 3 bits = 2^3 = 8 choix possibles | 3 bits = 2^3 = 8 choix possibles | 3 bits = 2^3 = 8 choix possibles

8 X 8 X 8 **512 droits possibles**

8 X 8 X 8 X 8 **4096 droits possibles**

Toutes les valeurs possibles des droits décimaux sur Linux

En résumé, en englobant les droits spéciaux, il y a jusqu'à 16 types de trois possibles pour les trois classes : u, o et g.

Ou bien alors 8 choix possibles avec les droits classiques.

u			g			o		
r	w	x	r	w	x	r	w	x
r	w	-	r	w	-	r	w	-
r	-	-	r	-	-	r	-	-
-	-	x	-	-	x	-	-	x
-	w	-	-	w	-	-	w	-
-	w	x	-	w	x	-	w	x
r	-	x	r	-	x	r	-	x
-	-	-	-	-	-	-	-	-
-	-	S	-	-	S	-	-	S
-	-	s	-	-	s	-	-	s
r	-	s	r	-	s	r	-	s
r	w	S	r	w	S	r	w	S
r	w	s	r	w	s	r	w	s
-	w	s	-	w	s	-	w	s
	w	S		w	S		w	S

Droits par défaut à la création

- **Les droits maximaux**
  - **Un fichier est créé avec des droits maximaux**

	Droits maximaux
fichier créé à partir d'un fichier source	les droits du fichier source
fichier créé à partir d'une redirection	666
répertoire	777
fichier créé par une application	définis par l'application

Droits par défaut à la création

- **La commande umask**
  - **A ces droits maximaux sont retranchés un masque de droits définis par la commande umask**

    umask valeur_du_masque_en_octal
  - **Par défaut la valeur du masque = 0022**

Exemple droits à la création

- **$ umask 0026**

- **$ >fic (redirection : droits maximaux=666)**

droits maximaux	:	- rw- rw- rw-
masque	:	- --- -w- rw-
droits de fic	:	- rw- r-- ---

- **$ mkdir rep (répertoire : droits maximaux=777)**

droits maximaux	:	- rwx rwx rwx
masque	:	- --- -w- rw-
droits de rep	:	- rwx r-x --x

Attributs des fichiers

- **Pour les systèmes de fichiers ext2 ou ext3 , 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…**

Attributs étendus

- Commandes
  - lsattr : lister les attributs d'un fichier
  - chattr : changer les attributs d'un fichier
    - chattr +a fic
  - L'option –R applique le droit à la sous-arborescence
    - chattr –R +i rep
- Ces attributs sont ignorés lors des copies et déplacements

- **Pratique des commandes**
  - **ls –l**
  - **chmod**
  - **umask**
  - **…**