Gestion des permissions et droits d’accès

Les droits d'accès

- Un système d'exploitation multi-utilisateur est conçu pour permettre à plusieurs utilisateurs d'utiliser l'ordinateur ''simutanément''. Dans notre système, on aura donc un grand nombre d’utilisateurs ayant éventuellement plusieurs fichiers.
- Le système d’exploitation permet aux utilisateurs d’utiliser des fichiers, tout en limitant les droits d'accès de chacun afin de garantir l'intégrité de leurs données.

Les droits d'accès

Les droits d'accès définissent :

- un fichier ou un répertoire appartient à un utilisateur,
- un fichier ou un répertoire appartient à un groupe d'utilisateurs.
- les actions des utilisateurs qu'ils peuvent effectuer sur les fichiers selon qu'ils sont :
  - propriétaire du fichier,
  - du même groupe que le propriétaire du fichier,
  - ou ni l'un ni l'autre. La gestion des permissions associées s'effectue sur chaque fichier (individuellement).

Les différents utilisateurs d’un fichier

- Par la propriété d'un fichier, on désigne à quel utilisateur appartient le fichier, qui le possède.
- À partir de cette possession (ou non), il sera ensuite possible de définir des permissions d'accès sur le fichier.
- Les permissions d'accès sur le fichier sont définies par le propriétaire du fichier
- Trois catégories d’utilisateurs d’un fichier :

1. **l'utilisateur** du fichier (**u**) : c’est le créateur du fichier.

2. **le groupe** du fichier (**g**). l’utilisateur est du même groupe que le créateur d'un fichier,

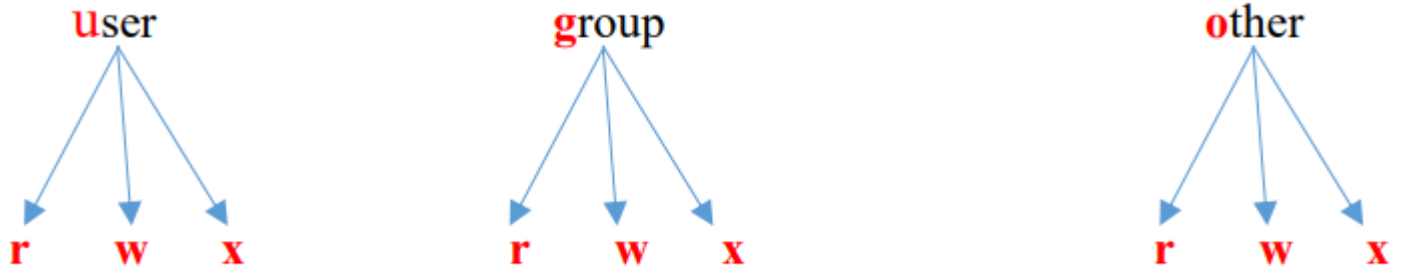
3. **les autres**, **other, le reste du monde** (**o**). ni propriétaire du fichier, ni membre du même groupe que le propriétaire du fichier.

Les permissions

- Un utilisateur quelconque (u, g ou o) peut avoir accès en lecture et écriture à un fichier, et n’a aucun accès sur un autre fichier par exemple. Trois types permissions :
- 1. **la lecture** d'un fichier (**r**) permet d’accéder au contenu d'un fichier :
  - écouter une piste audio, o visionner un film, lire un texte,
  - lister le contenu (ls) d’un répertoire.
- 2. **l'écriture** dans un fichier (**w**) permet de :
  - créer et modifier un fichier (corriger un texte et effectuer des mises à jour),
  - renommer ou supprimer un fichier dans un dossier ; etc.).
- 3. **l'exécution** d'un fichier (**x**) : permet :
  - d’exécuter des programmes (logiciels,scripts)
  - de faire d’le répertoire courant (cd)

Les permissions

- Pour chaque utilisateur (u, g ou o) sont définies ces trois permissions
  - le propriétaire (u) dispose ou non de la permission : r, w, x sur un fichier ;
  - le membre du groupe (g) dispose ou non de la permission : r, w, x sur un fichier ;
  - tous les autres utilisateurs (o) disposent ou non de la permission : r, w, x sur un fichier



Les permissions

- Les droits sont affichés (ls –l) par une série de 9 caractères, associé 3 par 3 (rwx rwx rwx) définissent les droits des 3 identités (u, g et o).
- Un dixième caractère indiquent le type de fichier.
  - \- : fichier ordinaire
  - d : répertoire (directory)
  - etc ..
- ex : - rwx rwx rwx

Chmod

- La commande **chmod** permet de **modifier l'attribution des droits d'un fichier ou d'un dossier**. Pour cela on ressort d'un côté les trois types d'utilisateurs : le user propriétaire u, le group propriétaire g et les others o, et d'un autre côté les trois droits read, write et execute.
- On donne a chmod les modifications que l'on souhaite apporter puis la liste des fichiers/dossiers auquel l'appliquer. Une modification de droit est décrite par:

  [**usertype**][**action**][**permissions**]

- **usertype** : est **u**, **g** ou **o** (ou a pour all)
- **action** : est **+ -** ou **=** pour ajouter, retirer ou fixer des permissions
- **Permissions :** est **r w** ou **x**

Chmod

- Alternativement, on attribue 3 nombres de 0 à 7 correspondant aux 3 types d'utilisateurs tel que :
  - r vaut : 4
  - w vaut : 2
  - x vaut : 1
- on somme les droits que l'on veut cumuler

Exemples :

- chmod 640 file donne les droits rw (4+2) à *user*, r (4) à *group* et aucun droit (0) à *others*.
- chmod 775 file donne les droits rwx (4+2+1) à *user* et *group* et rx (4+1) à *others*.

Chmod

- On peut aussi combiner plusieurs actions en même temps :

- chmod u+rwx,g+rx-w,o+r-wx monFile.txt :
  - user : lecture, écriture et exécution autorisées ;
  - groupe : lecture et exécution autorisées, écriture interdite,
  - autres : lecture autorisée, écriture et exécution interdites,

- chmod ugo+rwx monFile.txt : donne à tout le monde (propriétaire, groupe et autres) tous les droits sur monFile.txt.

- chmod a+rx monFile.txt : donne les droits de lecture et d'exécution à tout le monde (all).

- etc .

Umask

- La commande umask (user file-creation mode mask) spécifie les permissions qui doivent être supprimées des permissions de base lors de la création d'un nouveau fichier ou répertoire.

- Pour les fichiers, la permission de base est 666 (rw-rw-rw-).
- Pour les répertoires, la permission de base est 777 (rwxrwxrwx).

- Exemple:
- Nous allons définir le umask à 027. Cette valeur indique au système de supprimer les permissions suivantes :
- Utilisateur (propriétaire) : 0 - Aucune permission supprimée.
- Groupe : 2 - Permission d'écriture supprimée.
- Autres : 7 (4+2+1) - Permissions de lecture, d'écriture et d'exécution supprimées.
- Cela signifie que tout nouveau fichier créé aura les permissions 666 - 027 = 640 (rw-r-----), et tout nouveau répertoire aura les permissions 777 - 027 = 750 (rwxr-x---).

Chown

- La commande **chown** est utilisée pour modifier le propriétaire d'un fichier ou d'un répertoire. Elle nécessite des privilèges root (**sudo**) pour changer la propriété en faveur d'un autre utilisateur.
- chown user fichier → change le propriétaire.
- chown user:group fichier → change propriétaire + groupe.
- chown -R user:group dossier → applique récursivement à tous les fichiers et sous-dossiers.
- Nécessite **sudo** car seul root peut attribuer un fichier à un autre utilisateur.

Gestion de droits sous windows

- La gestion fine des permissions (lecture, écriture, exécution, héritage). On utilise **icacls**
  - Pour afficher les droits : icacls fichier.txt
  - Pour donner lecture, écriture, exécution à user: icacls fichier.txt /grant user:rwx
- La gestion des attributs simples (lecture seule, caché). On utilise **attrib**
  - attrib +r fichier.txt :met en lecture seule
  - attrib -r fichier.txt : enlève lecture seule
- Prendre possession d'un fichier/dossier. On utilise **takeown**
  - takeown /f C:\Users\Public\document.txt

  -> Le fichier document.txt devient la propriété de l'utilisateur courant (administrateur).

TP 1

1. Que signifient les droits **r**, **w** et **x** associés à un fichier ordinaire ?
2. Que signifient les droits **r**, **w** et **x** associés à un répertoire ?
3. Convertir les droits suivants en notation octale : **rwxr-xr--**
4. Convertir les droits suivants en notation symbolique : **640**
5. Quelle commande est utilisée pour attribuer le droit de lecture-écriture au propriétaire du fichier file.txt?
6. Avec la valeur umask de 112, quelle est la permission par défaut attribuée à un fichier nouvellement créé?

TP 2

1. Sous l'identité de l'utilisateur **user1** dans le répertoire */home/user1/chapitre7*.
2. Positionnez votre masque à la valeur **002**.
3. Créez le répertoire *docperso* et affectez-lui les droits **700**.
4. Allez dans le répertoire *docperso*, créez les fichiers *fica*, *ficb*, *ficc* et *ficd*, puis listez de façon détaillée le contenu du répertoire.
5. Créez le lien dur ("hard link") */tmp/lienfica* vers le fichier *fica*, et le lien symbolique ("soft link") */tmp/lienficb* vers le fichier *ficb*.
6. Retournez sous l'identité de l'utilisateur **user2** et essayez de lister le contenu du répertoire */home/user1/chapitre7/docperso*. Est-ce possible et pourquoi ?
7. Affichez le contenu du fichier */tmp/lienfica*. Est-ce possible et pourquoi ?
8. Affichez le contenu du fichier */tmp/lienficb*. Est-ce possible et pourquoi ?

TP 3

- Créer les utilisateurs et groupes suivants:

adduser userA

adduser userB

adduser userC

addgroup administrateurs

addgroup utilisateurs

usermod -G administrateurs userA

usermod -G administrateurs,utilisateurs userC

usermod -G utilisateurs userB

- Dans le dossier /home/userC créez :

1. un fichier nommé README que seul userC peut modifier, mais que tout le monde pourra lire
2. un dossier nommé admin accessible en lecture et écriture a toutes les personnes du groupe administrateurs, et à personne d'autre
3. un fichier dans le dossier admin nommé secret.txt que seul seuls les membres du groupe administrateurs pourront lire et modifier

TP 3

On commence par créer (en root) les dossiers et fichiers.

1. mkdir /home/userC/admin
2. touch /home/userC/README /home/userC/admin/secret.txt

Puis on attribue les bons propriétaires et groupes.

1. chown userC:userC /home/userC/README
2. chown userC:administrateurs /home/userC/admin
3. chown userC:administrateurs /home/userC/admin/secret.txt

Enfin on configure correctement les droits d'accès.

1. chmod 644 /home/userC/README
2. chmod 770 /home/userC/admin
3. chmod 660 /home/userC/admin/secret.txt

- On peut ensuite observer la bonne configuration des droits.