

Accès droits et attributs sous Linux

Ivan Noyer

Lycée Thiers

- la documentation UBUNTU

Propriétaire

Le *propriétaire* d'un fichier est l'utilisateur qui le possède. C'est souvent l'auteur du fichier mais pas toujours. Il y a trois degrés de possession d'un fichier :

l'utilisateur du fichier (**u**). C'est souvent le propriétaire. Remarquons qu'un fichier créé avec la commande **sudo** appartient à l'utilisateur **root**.

L'attribut de propriété d'un fichier prend donc 3 valeurs : **u,g,o**.

Propriétaire

Le *propriétaire* d'un fichier est l'utilisateur qui le possède. C'est souvent l'auteur du fichier mais pas toujours. Il y a trois degrés de possession d'un fichier :

l'*utilisateur* du fichier (**u**). C'est souvent le propriétaire. Remarquons qu'un fichier créé avec la commande **sudo** appartient à l'utilisateur **root**.

le *groupe propriétaire* du fichier (**g**). Si un utilisateur est membre d'un groupe qui possède la propriété d'un fichier, l'utilisateur aura aussi certaines permissions sur ce fichier.

L'attribut de propriété d'un fichier prend donc 3 valeurs : **u,g,o**.

Propriétaire

Le *propriétaire* d'un fichier est l'utilisateur qui le possède. C'est souvent l'auteur du fichier mais pas toujours. Il y a trois degrés de possession d'un fichier :

l'*utilisateur* du fichier (**u**). C'est souvent le propriétaire. Remarquons qu'un fichier créé avec la commande **sudo** appartient à l'utilisateur **root**.

le *groupe propriétaire* du fichier (**g**). Si un utilisateur est membre d'un groupe qui possède la propriété d'un fichier, l'utilisateur aura aussi certaines permissions sur ce fichier.

les *autres* (ou *other* (**o**)). Tous ceux qui ne sont pas dans le groupe propriétaire du fichier.

L'attribut de propriété d'un fichier prend donc 3 valeurs : **u,g,o**.

Permissions

Les *permissions* désignent ce que les diverses catégories d'utilisateurs ont l'autorisation d'effectuer sur un fichier donné (lecture, écriture, exécution). Il y a 3 catégories de permission : lecture (**r**), écriture (**w**) et eXécution (**x**).

Permissions pour un fichier régulier

lecture Nécessaire pour pouvoir accéder au *contenu* d'un fichier (écouter une piste audio, visionner un film, lire un texte).
Notation **r** (pour read, lire).

Permissions pour un fichier régulier

- lecture** Nécessaire pour pouvoir accéder au *contenu* d'un fichier (écouter une piste audio, visionner un film, lire un texte).
Notation **r** (pour read, lire).
- écriture** Nécessaire pour pouvoir apporter des modifications à un fichier (corriger un texte et enregistrer les changements ; effacer les *yeux rouges* dans une photo et enregistrer la correction etc.). Notation **w** (pour write, écrire).

Permissions pour un fichier régulier

- lecture** Nécessaire pour pouvoir accéder au *contenu* d'un fichier (écouter une piste audio, visionner un film, lire un texte).
Notation **r** (pour read, lire).
- écriture** Nécessaire pour pouvoir apporter des modifications à un fichier (corriger un texte et enregistrer les changements ; effacer les *yeux rouges* dans une photo et enregistrer la correction etc.). Notation **w** (pour write, écrire).
- exécution** Nécessaire pour les programmes, afin qu'ils puissent être exécutés. Cette permission est notée **x** (pour execute, exécuter).

Permissions pour un répertoire

lecture les droits en lecture permettent de lister le contenu d'un répertoire.

Permissions pour un répertoire

- lecture** les droits en lecture permettent de lister le contenu d'un répertoire.
- écriture** les droits en écriture signifient qu'on peut modifier le contenu donc ajouter, modifier, renommer ou supprimer un fichier dans un dossier.

Permissions pour un répertoire

- lecture** les droits en lecture permettent de lister le contenu d'un répertoire.
- écriture** les droits en écriture signifient qu'on peut modifier le contenu donc ajouter, modifier, renommer ou supprimer un fichier dans un dossier.
- exécution** Pour un répertoire, la permission **x** permet d'en faire le répertoire courant et donc d'y accéder par **cd**.

Exemple

Supposons que l'utilisateur **toto** dispose des droits de lecture et d'exécution sur le répertoire **Foo** mais pas du droit d'écriture. Alors **toto** peut lister le contenu de **Foo** et se rendre dans **Foo** exécuter les programmes contenus dans **Foo** et lire leur contenu. Mais il ne peut pas ajouter/supprimer de fichiers dans **Foo**.

9 caractères pour définir tous les droits

- Pour chacune des trois catégories d'utilisateurs (propriétaire, membres du groupe propriétaire et reste du monde) sont définies ces trois permissions :

9 caractères pour définir tous les droits

- Pour chacune des trois catégories d'utilisateurs (propriétaire, membres du groupe propriétaire et reste du monde) sont définies ces trois permissions :
 - le propriétaire dispose ou non de la permission de lecture, d'écriture et d'exécution sur un fichier,

9 caractères pour définir tous les droits

- Pour chacune des trois catégories d'utilisateurs (propriétaire, membres du groupe propriétaire et reste du monde) sont définies ces trois permissions :
 - le propriétaire dispose ou non de la permission de lecture, d'écriture et d'exécution sur un fichier,
 - le membre du groupe propriétaire dispose ou non de la permission de lecture, d'écriture et d'exécution sur un fichier,

9 caractères pour définir tous les droits

- Pour chacune des trois catégories d'utilisateurs (propriétaire, membres du groupe propriétaire et reste du monde) sont définies ces trois permissions :
 - le propriétaire dispose ou non de la permission de lecture, d'écriture et d'exécution sur un fichier,
 - le membre du groupe propriétaire dispose ou non de la permission de lecture, d'écriture et d'exécution sur un fichier,
 - tous les autres utilisateurs disposent ou non de la permission de lecture, d'écriture et d'exécution sur un fichier.

9 caractères pour définir tous les droits

- Pour chacune des trois catégories d'utilisateurs (propriétaire, membres du groupe propriétaire et reste du monde) sont définies ces trois permissions :
 - le propriétaire dispose ou non de la permission de lecture, d'écriture et d'exécution sur un fichier,
 - le membre du groupe propriétaire dispose ou non de la permission de lecture, d'écriture et d'exécution sur un fichier,
 - tous les autres utilisateurs disposent ou non de la permission de lecture, d'écriture et d'exécution sur un fichier.
- Les droits sont donc affichés par une série de 9 caractères, associés 3 par 3 (**rwX rwX rwX**) qui définissent les droits des 3 identités (**u, g, o** dans cet ordre).

9 caractères pour définir tous les droits

- Pour chacune des trois catégories d'utilisateurs (propriétaire, membres du groupe propriétaire et reste du monde) sont définies ces trois permissions :
 - le propriétaire dispose ou non de la permission de lecture, d'écriture et d'exécution sur un fichier,
 - le membre du groupe propriétaire dispose ou non de la permission de lecture, d'écriture et d'exécution sur un fichier,
 - tous les autres utilisateurs disposent ou non de la permission de lecture, d'écriture et d'exécution sur un fichier.
- Les droits sont donc affichés par une série de 9 caractères, associés 3 par 3 (**rwX rwX rwX**) qui définissent les droits des 3 identités (**u, g, o** dans cet ordre).
- il y a encore deux droits spéciaux **s,t** (hors programme). aborde pas ici.

Slide hors programme

- Le bit **Set-User-ID** permet à un utilisateur d'exécuter le programme avec les droits du propriétaire, c'est ainsi que sudo nous permet d'exécuter des commandes en "root"

```
$ ls -l /usr/bin | grep sudo  
-rwsr-xr-x 1 root root 166056 avril 4 2023 sudo
```

Slide hors programme

- Le bit **Set-User-ID** permet à un utilisateur d'exécuter le programme avec les droits du propriétaire, c'est ainsi que sudo nous permet d'exécuter des commandes en "root"

```
$ ls -l /usr/bin | grep sudo  
-rwsr-xr-x 1 root root 166056 avril 4 2023 sudo
```

- le bit **Set-Group-ID** agit comme **Set-User-ID** mais pour le groupe

```
$ ls -l /usr/bin | grep ssh-agent  
-rwxr-sr-x 1 root ssh 350504 janv. 2 18:13 ssh-agent
```

Droits spéciaux

Slide hors programme

- Le bit « restriction de suppression » ou **Sticky** permet quant à lui de restreindre la suppression d'un fichier ou répertoire à son seul propriétaire. C'est le cas du répertoire **/tmp** :

```
$ ls -ld /t*/  
drwxrwxrwt 21 root root 36864 janv. 24 11:46 /tmp/
```

Droits spéciaux

Slide hors programme

- Le bit « restriction de suppression » ou **Sticky** permet quant à lui de restreindre la suppression d'un fichier ou répertoire à son seul propriétaire. C'est le cas du répertoire **/tmp** :

```
$ ls -ld /t*/  
drwxrwxrwt 21 root root 36864 janv. 24 11:46 /tmp/
```

- Le **t** au lieu du **x** pour les autres utilisateurs nous informe que ce répertoire ne peut être supprimé que par l'utilisateur root

Droits spéciaux

Slide hors programme

- Le bit « restriction de suppression » ou **Sticky** permet quant à lui de restreindre la suppression d'un fichier ou répertoire à son seul propriétaire. C'est le cas du répertoire **/tmp** :

```
$ ls -ld /t*/  
drwxrwxrwt 21 root root 36864 janv. 24 11:46 /tmp/
```

- Le **t** au lieu du **x** pour les autres utilisateurs nous informe que ce répertoire ne peut être supprimé que par l'utilisateur root
- On peut donc en mode octal accorder ces droits spéciaux en ajoutant une quatrième information. Pour activer le sticky bit et le Set-GroupID sur le script **renomme_mes_photos.sh**, on entre :

```
$ chmod 3777 renomme_mes_photos.sh
```

Notations octales

Chaque groupe de trois caractères de permissions (comme **rW-** ou **-X**) est représenté par un chiffre entre 0 et 7 et chaque droit correspond à une valeur :

- **r** (read) = 4

Notations octales

Chaque groupe de trois caractères de permissions (comme **rW-** ou **-X**) est représenté par un chiffre entre 0 et 7 et chaque droit correspond à une valeur :

- **r** (read) = 4
- **w** (write) = 2

Notations octales

Chaque groupe de trois caractères de permissions (comme **rW-** ou **-X**) est représenté par un chiffre entre 0 et 7 et chaque droit correspond à une valeur :

- **r** (read) = 4
- **w** (write) = 2
- **x** (execute) = 1

Notations octales

Chaque groupe de trois caractères de permissions (comme **rW-** ou **-X**) est représenté par un chiffre entre 0 et 7 et chaque droit correspond à une valeur :

- **r** (read) = 4
- **w** (write) = 2
- **x** (execute) = 1
- **-** = 0

Notations octales

Chaque groupe de trois caractères de permissions (comme **rW-** ou **-X**) est représenté par un chiffre entre 0 et 7 et chaque droit correspond à une valeur :

- **r** (read) = 4
- **w** (write) = 2
- **x** (execute) = 1
- **-** = 0
- Par exemple pour le triplet :

Notations octales

Chaque groupe de trois caractères de permissions (comme **rw-** ou **-x**) est représenté par un chiffre entre 0 et 7 et chaque droit correspond à une valeur :

- **r** (read) = 4
- **w** (write) = 2
- **x** (execute) = 1
- **-** = 0
- Par exemple pour le triplet :
 - **rwX**, on a : $4+2+1 = 7$

Notations octales

Chaque groupe de trois caractères de permissions (comme **rw-** ou **-x**) est représenté par un chiffre entre 0 et 7 et chaque droit correspond à une valeur :

- **r** (read) = 4
- **w** (write) = 2
- **x** (execute) = 1
- **-** = 0
- Par exemple pour le triplet :
 - **rwX**, on a : $4+2+1 = 7$
 - **rw-**, on a $4+2+0=6$

Notations octales

Chaque groupe de trois caractères de permissions (comme **rw-** ou **-x**) est représenté par un chiffre entre 0 et 7 et chaque droit correspond à une valeur :

- **r** (read) = 4
- **w** (write) = 2
- **x** (execute) = 1
- **-** = 0
- Par exemple pour le triplet :
 - **rwX**, on a : $4+2+1 = 7$
 - **rw-**, on a $4+2+0=6$
 - **r-**, on a : $4+0+0=4$

Notations octales

On obtient ainsi toutes les combinaisons :

- 0 - - - (aucun droit)
- 1 - - **x** (exécution)
- 2 - **w** - (écriture)
- 3 - **w x** (écriture et exécution)
- 4 **r** - - (lecture seule)
- 5 **r - x** (lecture et exécution)
- 6 **r w** - (lecture et écriture)
- 7 **r w x** (lecture, écriture et exécution)

Exemple

Considérons un répertoire dont les attributs sont **drwxr-x---** (« **d** » pour répertoire) :

- **rwX** : $(4+2+1)$ soit 7

Exemple

Considérons un répertoire dont les attributs sont **drwxr-x---** (« **d** » pour répertoire) :

- **rwX** : $(4+2+1)$ soit 7
- **r-X** : $(4+0+1)$ soit 5

Exemple

Considérons un répertoire dont les attributs sont **drwxr-x---** (« **d** » pour répertoire) :

- **rwX** : $(4+2+1)$ soit 7
- **r-X** : $(4+0+1)$ soit 5
- **---** : $(0+0+0)$ soit 0

Exemple

Considérons un répertoire dont les attributs sont **drwxr-x---** (« **d** » pour répertoire) :

- **rwX** : $(4+2+1)$ soit 7
- **r-x** : $(4+0+1)$ soit 5
- **---** : $(0+0+0)$ soit 0
- les droits de ce répertoire sont donc résumés par le nombre **750**

Changer les droits d'un fichier

L'outil **chmod** (de « change mode » en anglais) permet de modifier les permissions sur un fichier. Il s'emploie de deux façons

- soit en précisant les permissions de manière octale : on entre alors une suite de trois chiffres entre 0 et 7 pour désigner l'utilisateur, son groupe et le reste du monde.

Changer les droits d'un fichier

L'outil **chmod** (de « change mode » en anglais) permet de modifier les permissions sur un fichier. Il s'emploie de deux façons

- soit en précisant les permissions de manière octale : on entre alors une suite de trois chiffres entre 0 et 7 pour désigner l'utilisateur, son groupe et le reste du monde.
- soit en précisant à qui s'adresse le changement et en ajoutant ou retirant des permissions

Changer les droits d'un fichier

En gérant les droits séparément, on choisit dans l'ordre :

Qui est concerné cela peut être

Changer les droits d'un fichier

En gérant les droits séparément, on choisit dans l'ordre :

Qui est concerné cela peut être

- **u** (user, utilisateur) représente la catégorie "propriétaire",

Changer les droits d'un fichier

En gérant les droits séparément, on choisit dans l'ordre :

Qui est concerné cela peut être

- **u** (user, utilisateur) représente la catégorie "propriétaire",
- **g** (group, groupe) représente la catégorie "groupe propriétaire",

Changer les droits d'un fichier

En gérant les droits séparément, on choisit dans l'ordre :

Qui est concerné cela peut être

- **u** (user, utilisateur) représente la catégorie "propriétaire",
- **g** (group, groupe) représente la catégorie "groupe propriétaire",
- **o** (others, autres) représente la catégorie "reste du monde",

Changer les droits d'un fichier

En gérant les droits séparément, on choisit dans l'ordre :

Qui est concerné cela peut être

- **u** (user, utilisateur) représente la catégorie "propriétaire",
- **g** (group, groupe) représente la catégorie "groupe propriétaire",
- **o** (others, autres) représente la catégorie "reste du monde",
- **a** (all, tous) représente l'ensemble des trois catégories.

Changer les droits d'un fichier

En gérant les droits séparément, on choisit dans l'ordre :

Qui est concerné cela peut être

- **u** (user, utilisateur) représente la catégorie "propriétaire",
- **g** (group, groupe) représente la catégorie "groupe propriétaire",
- **o** (others, autres) représente la catégorie "reste du monde",
- **a** (all, tous) représente l'ensemble des trois catégories.

la modification : ajouter ou supprimer

Changer les droits d'un fichier

En gérant les droits séparément, on choisit dans l'ordre :

Qui est concerné cela peut être

- **u** (user, utilisateur) représente la catégorie "propriétaire",
- **g** (group, groupe) représente la catégorie "groupe propriétaire",
- **o** (others, autres) représente la catégorie "reste du monde",
- **a** (all, tous) représente l'ensemble des trois catégories.

la modification : ajouter ou supprimer

- **+** pour ajouter un droit

Changer les droits d'un fichier

En gérant les droits séparément, on choisit dans l'ordre :

Qui est concerné cela peut être

- **u** (user, utilisateur) représente la catégorie "propriétaire",
- **g** (group, groupe) représente la catégorie "groupe propriétaire",
- **o** (others, autres) représente la catégorie "reste du monde",
- **a** (all, tous) représente l'ensemble des trois catégories.

la modification : ajouter ou supprimer

- **+** pour ajouter un droit
- **-** pour supprimer un droit

Changer les droits d'un fichier

En gérant les droits séparément, on choisit dans l'ordre :

Qui est concerné cela peut être

- **u** (user, utilisateur) représente la catégorie "propriétaire",
- **g** (group, groupe) représente la catégorie "groupe propriétaire",
- **o** (others, autres) représente la catégorie "reste du monde",
- **a** (all, tous) représente l'ensemble des trois catégories.

la modification : ajouter ou supprimer

- **+** pour ajouter un droit
- **-** pour supprimer un droit
- **=** pour écraser les 3 droits et en mettre de nouveaux

Changer les droits d'un fichier

En gérant les droits séparément, on choisit dans l'ordre :

Qui est concerné cela peut être

- **u** (user, utilisateur) représente la catégorie "propriétaire",
- **g** (group, groupe) représente la catégorie "groupe propriétaire",
- **o** (others, autres) représente la catégorie "reste du monde",
- **a** (all, tous) représente l'ensemble des trois catégories.

la modification : ajouter ou supprimer

- **+** pour ajouter un droit
- **-** pour supprimer un droit
- **=** pour écraser les 3 droits et en mettre de nouveaux

le droit modifié **r**, **w** ou **x**

Exemples

- **chmod o=r myfile** donne les droits de lecture (et pas au groupe ni au propriétaire) aux autres utilisateurs sans se préoccuper des anciens droits qu'avait **o**.

Exemples

- **chmod o=r myfile** donne les droits de lecture (et pas au groupe ni au propriétaire) aux autres utilisateurs sans se préoccuper des anciens droits qu'avait **o**.
- **chmod 644 myfile** : droit utilisé traditionnellement sur les fichiers. Donne au propriétaire les droits de modification et lecture, aux membres du groupe et aux autres uniquement les droits de lecture.

Exemples

- **chmod o=r myfile** donne les droits de lecture (et pas au groupe ni au propriétaire) aux autres utilisateurs sans se préoccuper des anciens droits qu'avait **o**.
- **chmod 644 myfile** : droit utilisé traditionnellement sur les fichiers. Donne au propriétaire les droits de modification et lecture, aux membres du groupe et aux autres uniquement les droits de lecture.
- **chmod 755 mydir** : droit utilisé traditionnellement sur les répertoires. Donne au propriétaire tous les droits, aux membres du groupe et aux autres les droits de lecture (par **ls** et d'accès (par **cd**)).

Exemples

- **chmod o=r myfile** donne les droits de lecture (et pas au groupe ni au propriétaire) aux autres utilisateurs sans se préoccuper des anciens droits qu'avait **o**.
- **chmod 644 myfile** : droit utilisé traditionnellement sur les fichiers. Donne au propriétaire les droits de modification et lecture, aux membres du groupe et aux autres uniquement les droits de lecture.
- **chmod 755 mydir** : droit utilisé traditionnellement sur les répertoires. Donne au propriétaire tous les droits, aux membres du groupe et aux autres les droits de lecture (par **ls** et d'accès (par **cd**)).
- **chmod u+x myfile** donne au propriétaire les droits d'exécutions sur **myfile**. **chmod a+x myfile** donne à tout le monde les droits d'exécution.

Exemples

- **chmod o=r myfile** donne les droits de lecture (et pas au groupe ni au propriétaire) aux autres utilisateurs sans se préoccuper des anciens droits qu'avait **o**.
- **chmod 644 myfile** : droit utilisé traditionnellement sur les fichiers. Donne au propriétaire les droits de modification et lecture, aux membres du groupe et aux autres uniquement les droits de lecture.
- **chmod 755 mydir** : droit utilisé traditionnellement sur les répertoires. Donne au propriétaire tous les droits, aux membres du groupe et aux autres les droits de lecture (par **ls** et d'accès (par **cd**)).
- **chmod u+x myfile** donne au propriétaire les droits d'exécutions sur **myfile**. **chmod a+x myfile** donne à tout le monde les droits d'exécution.
- **chmod -R a+rx mydir** donne à tous les utilisateurs les droits en lecture et en exécution à tout ce que contient le dossier **mydir**.

Exercice

- 1 Écrire un fichier de script bash **myscript.sh** qui affiche « coucou » à l'écran (la première ligne d'un tel fichier est `#!/bin/bash`). Cela doit se faire en une seule ligne de commande sans ouvrir vim, nano, gedit ou emacs.
- 2 L'exécuter avec `.\myscript.sh`. Que constate-t-on ?
- 3 Reemédier à ce problème.