

DEV1 – ENVL – Laboratoires d'environnement

TD 2 – Les permissions

Table des matières

1 Se documenter

1.1 Les options

La plupart des commandes possèdent des **options**.

Les options

Une **option** **modifie le sens** d'une commande ;

- ▷ Elle commence par le signe - suivi d'une seule lettre ;
- ▷ ou encore par le double tiret -- suivi d'un nom d'option.
- ▷ Elle est placée n'importe où après le nom de la commande.

Experience 1

Expérimenter les options

- ✍ Tapez la commande `ls -l`. Vous constatez que le résultat obtenu est beaucoup plus verbeux que celui obtenu sans l'option.
- ✍ Placez vous dans le dossier `td1` et tapez la commande `cat -n test`. L'option demande de numéroté les lignes
- ✍ Essayez `cat --number test`, la version *longue* équivalente à la précédente.

1.2 Recherche d'informations

Non seulement, il y a beaucoup de commandes à connaître mais, en plus, chacune dispose d'une multitude d'options. Impossible de tout retenir ! Comment faire ?

Information

- ▷ `nomCommande --help` affiche une aide succincte sur la commande.
- ▷ `man nomCommande` affiche une documentation plus complète. (*q* pour quitter).

Si vous ne connaissez pas le nom de la commande, consultez les documents que l'on met à votre disposition, notamment le *Linux Cheat Sheet*.

Exercice 1

Trouver la bonne option

La commande `ls -l` affiche le contenu du dossier en format *long*. La 5^e colonne donne la taille du fichier (en octets). Lorsque les nombres sont grands, ce n'est pas très lisible. Trouvez l'option qui permet d'afficher cette taille sous un format plus lisible.

De grâce, **cherchez** la réponse, ne la **demandez pas** à votre voisin. Le but de cet exercice n'est pas de connaître l'option (elle n'est pas si utile que ça) mais d'apprendre à trouver soi-même l'information.

Exercice 2

Trouver la bonne commande

Quelle commande permet de

- ▷ *nettoyer* l'écran ?
- ▷ afficher la date et l'heure ?

2 Les permissions

Comme Linux est un système partagé, il est important de parler de sécurité. Ici, on va se concentrer sur la sécurité au niveau des fichiers et répondre aux questions suivantes :

- ▷ À qui **appartient** un fichier ?
- ▷ **Qui peut faire** quoi avec un fichier ?

Mais pour commencer il faut d'abord comprendre la notion de *propriétaire* et celle de *groupe*.

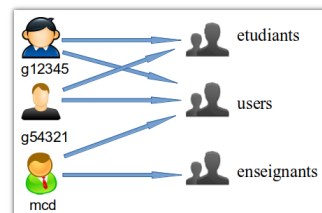
2.1 Les groupes d'utilisateurs

Les groupes

Les utilisateurs d'un système Linux sont placés dans des **groupes**. Un groupe contient un ou plusieurs utilisateur(s) et un utilisateur appartient à un ou plusieurs groupe(s).

Exemple : Dans l'exemple ci-contre on peut constater que :

- ▷ l'utilisateur `g12345` appartient aux groupes `users` et `etudiants` ;
- ▷ l'utilisateur `g54321` appartient aux mêmes groupes ;
- ▷ l'utilisateur `mcd` appartient aux groupes `users` et `enseignants`.



À quoi servent les groupes ? À gérer les permissions. On va pouvoir dire un truc du genre : « Seuls les professeurs peuvent voir le contenu de ce fichier ».

Visualiser les groupes

La commande **groups** renseigne sur les groupes.

- ▷ `groups` : affiche les groupes de l'utilisateur.
- ▷ `groups loginUtilisateur` : affiche les groupes de l'utilisateur donné.

Le premier groupe de la liste est le **groupe principal**. C'est celui qui sera utilisé par défaut (par exemple, lorsqu'on crée un fichier).

Exercice 3

Visualiser les groupes

- ▷ Visualiser les groupes auxquels vous appartenez.
- ▷ Quel est votre groupe principal ?
- ▷ Quels sont les groupes auxquels appartient votre professeur ?
- ▷ Avez-vous un groupe en commun avec lui ?
- ▷ Quel(s) groupe(s) Linux avez-vous en commun avec les autres étudiants de votre groupe ÉSI ?

Les groupes qui existent concrètement sur une machine sont définis par l'administrateur ^a de la machine. Sur `linux1`, il y a 3 groupes pour les utilisateurs.

- ▷ **users** : tous les utilisateurs sont dans ce groupe
- ▷ **etudiants** : tous les étudiants sont dans ce groupe
- ▷ **enseignants** : tous les professeurs sont dans ce groupe

^a. L'administrateur est la personne qui installe un système d'exploitation et le gère au quotidien : installation de logiciels, gestion des comptes utilisateurs et des groupes, sauvegardes, gestion des panne... Sur Linux, on dit aussi le « super user », le « compte root » ou le « root ».

Attention, on remarque que vous confondez souvent la notion de **groupe d'étudiants** à l'école et les **groupes sur Linux**. C'est un même mot qui recouvre 2 concepts complètement différents.

2.2 Propriétaire d'un fichier

« La propriété est un piège : ce que nous croyons posséder nous possède. » – Alphonse Karr

« La propriété c'est le vol. » – Pierre Joseph Proudhon

Propriétaire

Chaque fichier/dossier **appartient** à une personne, son **propriétaire**.

Pour visualiser le propriétaire d'un fichier, on peut utiliser la commande `ls -l`.

```
g12345@linux1:~> ls -l
total 8
drwxrwxr-x 2 g12345 users 4096 mai  6 23:46 dev1
-rw-rw-r-- 1 g12345 users  42 mai  6 23:46 welcome
```

Le propriétaire

On pourra donner des permissions spécifiques au propriétaire du fichier.

Exercice 4

Propriétaire

Visualisez le propriétaire des fichiers de votre dossier personnel.

FAQ

Les fichiers dans mon dossier personnel ne sont pas automatiquement à moi ?

Non. En pratique c'est généralement le cas, mais on peut très bien trouver dans un dossier personnel un fichier qui appartient à quelqu'un d'autre.

2.3 Groupes d'un fichier

Groupe d'un fichier

Chaque fichier/dossier **appartient** à un et un seul groupe.

```
g12345@linux1:~> ls -l
total 8
drwxrwxr-x 2 g12345 users 4096 mai  6 23:46 dev1
-rw-rw-r-- 1 g12345 users  42 mai  6 23:46 welcome
```

Le groupe du fichier

Ces groupes sont les mêmes que ceux utilisés pour grouper les utilisateurs. On pourra donner des permissions spécifiques à toutes les personnes qui sont dans le même groupe que le fichier.

Changer le groupe

`chgrp nouveauGroupe nomFichier` place le fichier indiqué dans le groupe donné.

Par défaut, un fichier est placé dans le groupe principal de celui qui le crée.

Exercice 5

Groupe

- ▷ Visualisez vos fichiers et déterminez à quel groupe ils appartiennent.
- ▷ Créez un fichier de test et modifiez le groupe auquel il appartient.

2.4 Les 3 catégories de personnes

Ce qu'on a déjà vu

Un fichier a un propriétaire et appartient à un groupe.

Restez concentré !

D'expérience nous constatons que vous avez du mal avec les notions qui vont suivre. Restez concentrés et vérifiez que vous avez bien compris.

Lorsque nous allons donner des permissions sur un fichier/dossier, nous allons pouvoir donner des droits différents à trois catégories de personnes : le propriétaire, les utilisateurs qui sont dans le même groupe que le fichier et tous les « autres » (ceux qui ne sont pas dans les deux premières catégories).

```
g12345@linux1:~> ls -l
total 8
drwxrwxr-x 2 g12345 users 4096 mai  6 23:46 dev1
-rw-rw-r-- 1 g12345 users  42 mai  6 23:46 welcome
```

Les permissions du groupe

Les permissions du propriétaire

Les permissions pour tous les « autres »

2.5 Les permissions sur un fichier

Nous avons vu que nous pouvons spécifier des permissions différentes à trois catégories de personnes. Bien ! Mais quelles permissions pouvons-nous donner ? À quoi correspondent les lettres **r**, **w** et **x** vues dans la capture d'écran précédente ? Commençons par les permissions pour un fichier.

Les permissions sur un fichier

Il existe 3 types de permissions pour un fichier : en lecture, en écriture et en exécution.

r (pour *Read*, la lecture)

Avec ce droit, un utilisateur peut *lire* un fichier, il peut en voir le contenu. Par exemple, avec `cat`.

w (pour *Write*, l'écriture)

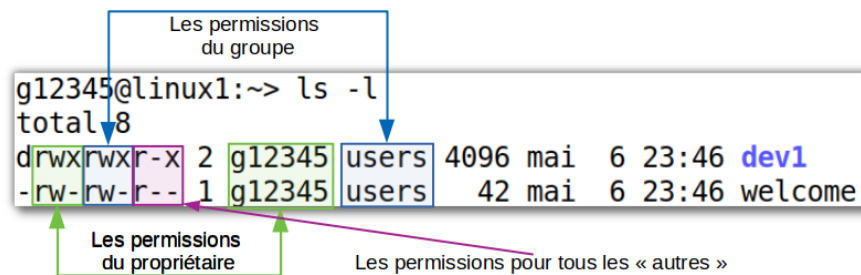
Un utilisateur a le droit en écriture sur un fichier, il peut en modifier le contenu. Par exemple, avec `nano`.

x (pour *eXecute*, l'exécution)

Cette permission concerne les exécutables. Un exécutable est un fichier qui contient un programme en langage machine directement exécutable par le processeur. Par exemple, il existe quelque part sur le disque un fichier `nano` qui contient le programme de la commande `nano`.

Les permissions sont toujours placées dans le même ordre (**rw**x) et un tiret (-) indique que la permission n'est pas accordée.

Exemple. Reprenons la capture d'écran précédente et interprétons les permissions sur le fichier `welcome`.



On y lit que :

- ▷ Les permissions pour le propriétaire sont **rw-**. Donc, (`g12345`) peut voir (**r**) le contenu et modifier (**w**) le fichier `welcome`. Par contre, il ne peut pas l'exécuter.
- ▷ Idem pour les utilisateurs qui sont dans le groupe `users`.
- ▷ Les autres utilisateurs (ceux qui ne sont ni le propriétaire, ni un utilisateur du groupe `users`) peuvent lire le fichier mais pas le modifier ni l'exécuter.

Exercice 6

Déterminez les bonnes permissions

Imaginons que vous écrivez un programme JAVA sur `linux1`. Vous ne voulez pas que quelqu'un copie sur vous. Par contre, vous désirez qu'il puisse exécuter la version compilée. Quelles permissions proposez-vous pour les fichiers :

- ▷ `.java`?
- ▷ `.class`?

Experience 2

Permissions par défaut

Quelle est la situation de départ quand on crée un fichier ? Faisons une petite expérience pour le découvrir.

- ✍ Si ce n'est pas encore fait, créez un dossier `dev1/td2`.

✍ Créez-y un fichier vide.

✍ Demandez les détails du fichier (propriétaire, groupe, permission)

On constate qu'un nouveau fichier appartient à celui qui l'a créé (on s'en doute) et au groupe principal du créateur. Il y a aussi des permissions par défaut (plutôt permissives dans notre cas).

2.6 Modifier les permissions

Faisons le point !

Nous savons qu'un fichier a un propriétaire et un groupe. Nous savons que nous allons pouvoir donner des permissions différentes au propriétaire, au membres du groupe et à tous les autres utilisateurs. Nous avons aussi appris à comprendre les permissions données sur un fichier.

Très bien mais comment modifier ces permissions ?

Modifier les permissions

`chmod permissions fichier` modifie les permissions du fichier.

Il existe deux façons d'indiquer les permissions : via des nombres ou via des lettres.

2.6.1 Donner les permissions avec des nombres

Pour indiquer les permissions avec un nombre, on considère que $r = 4$, $w = 2$ et $x = 1$. Et on additionne les valeurs ainsi obtenues.

	lettres	nombre	signification
Exemples :	<code>r--</code>	4 (4+0+0)	droit en lecture uniquement
	<code>rw-</code>	6 (4+2+0)	droit en lecture et en écriture
	<code>--x</code>	1 (0+0+1)	droit d'exécution uniquement

On combine alors les nombres obtenus pour les 3 sortes d'utilisateurs.

Exemple : `chmod 640 welcome` donne au propriétaire le droit en lecture/écriture, aux utilisateurs dans le même groupe que le fichier le droit en lecture uniquement et aucun droit aux autres.

Exercice 7

Convertir les permissions en nombres

Reprenez les permissions affichées dans la capture d'écran ci-dessous et exprimez-les avec un nombre de 3 chiffres.

```
-> ls -l
total 20
drwxr-xr-x 2 g32671 users 1024 mai 29 14:27 bin
drwxr-xr-x 2 g32671 users 1024 oct 21 2008 cours
drwx----- 2 g32671 users 1024 mai 29 14:27 Documents
drwxr-xr-x 3 g32671 users 1024 nov 20 2008 evaluation
-rw-r--r-- 1 g32671 users 417 oct 21 2008 Max.java
-rw-r--r-- 1 g32671 users 418 oct 21 2008 Max.java~
-rw-r--r-- 1 g32671 users 0 oct 21 2008 Maxnombre.java
```

Exercice 8

Modifier les permissions

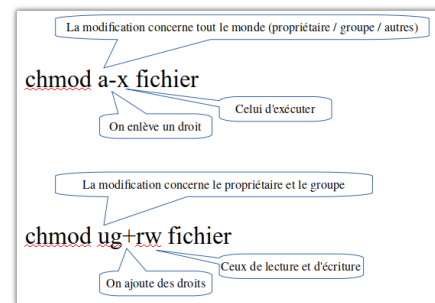
Créez un fichier `dev1/td2/bro1` contenant un petit texte de votre choix. Faites en sorte qu'il soit lisible par tout le monde mais modifiable uniquement par vous.

2.6.2 Donner les permissions avec des lettres

Avec la notation précédente, on spécifie toutes les permissions. Ici, on va aussi pouvoir indiquer uniquement ce qui doit changer : ajouter/enlever un droit sans toucher aux autres droits.

Le format de la commande est assez complexe.

- ▷ On indique d'abord les utilisateurs concernés par la modification des droits :
 - ▷ « u » pour le propriétaire (user),
 - ▷ « g » pour le groupe,
 - ▷ « o » pour les autres (other) et
 - ▷ « a » pour tout le monde (all)
- ▷ On indique ensuite si on veut ajouter (« + ») ou enlever (« - ») un droit
- ▷ On indique enfin quel(s) droit(s) on ajoute ou enlève (« r », « w » et/ou « x »)



Consultez à la page de manuel pour découvrir toutes les possibilités.

Exercice 9

Modifier les permissions

Créez un fichier `dev1/td2/bro12` contenant un petit texte de votre choix. Faites en sorte qu'il soit lisible par tout le monde mais modifiable uniquement par vous.

2.7 Différencier les étudiants des enseignants

Nous n'allons pas ici introduire un concept nouveau mais appliquer tout ce qui a été vu dans un exemple qui, par expérience, vous pose souvent un problème.

Tutoriel 1

Droit de lecture pour les enseignant uniquement

Supposons que vous avez un fichier qui ne doit être lu que par vous-même et les enseignants (mais pas les autres étudiants).

Comment donner des droits différents aux enseignants et aux étudiants ? La seule solution passe par une modification du groupe du fichier. S'il appartient au groupe **etudiants** par exemple et plus au groupe **users**, les permissions de groupe s'appliqueront aux étudiants et les permissions des *autres* concerneront les enseignants ¹

- ✍ Entrez `cd /dev1/td2` pour vous placer dans le dossier indiqué.
- ✍ Entrez `touch examen` pour créer le fichier (vide) indiqué.
- ✍ Entrez `chgrp etudiants examen` pour changer son groupe.
- ✍ Entrez `ls -l` pour visualiser la modification.

1. Pour être plus précis, tout utilisateur qui n'est pas dans le groupe **etudiants**.

✍ Entrez `chmod 604 examen` pour changer les permissions associées au fichier.

✍ Entrez `ls -l` pour visualiser le résultat final.

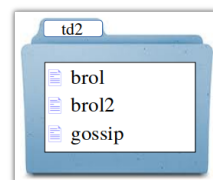
Exercice 10 Un fichier réservé aux étudiants

Créez un fichier `dev1/td2/gossip` pour partager des rumeurs entre élèves. On voudrait que tous les étudiants puissent le lire et le modifier mais que les enseignants ne puissent ni le lire, ni l'écrire.

2.8 Les permissions sur les dossiers

Tout comme pour les fichiers, on peut donner des permissions au dossier. Ce qui change, c'est la signification des permissions. Que veut dire « exécuter » un dossier par exemple ?

Pour bien comprendre, prenons une image. Imaginons qu'un dossier possède sur sa « couverture » une liste de son contenu. Par exemple, on voit ci-contre que le dossier `td2` contient 3 fichiers.



Permissions sur un dossier

Explicitons à présent le sens de chaque droit pour un dossier :

r (lecture)

On a le droit de lire ce qui est écrit sur la « couverture » du dossier. On peut par exemple faire `ls td2`.

w (écriture)

On peut modifier la « couverture » du dossier. On peut ainsi effacer un fichier (`rm td1/bro12`) ou en créer un (`touch td1/bro13`).

x (ouverture)

On peut « ouvrir » le dossier / entrer dedans. On peut ainsi en faire son dossier courant (`cd td2`) ou le traverser dans un chemin (`cat td2/gossip`).

Pour effacer un fichier il ne faut pas de droit en écriture sur le fichier mais bien sur le dossier qui le contient.

Exercice 11 Les permissions sur un dossier (I)

Dans un dossier `dir1` dans votre dossier `td2`. Dans ce dossier, créez un fichier `file`. Faites en sorte que tout le monde puisse voir quels fichiers se trouvent dans `dir1` mais sans pouvoir lire le contenu de ces fichiers.

Exercice 12 Les permissions sur un dossier (II)

Dans un dossier `dir2` dans votre dossier `td2`. Dans ce dossier, créez un fichier `file`. Faites en sorte que tout le monde puisse modifier ce fichier mais sans pouvoir le supprimer.

2.9 Les derniers détails

Voyons quelques détails mis de coté pendant notre apprentissage.

```
g12345@linux1:~> ls -l
total 8
drwxrwxr-x 2 g12345 users 4096 mai 6 23:46 dev1
-rw-rw-r-- 1 g12345 users 42 mai 6 23:46 welcome
```

FAQ

Vous dites que dans un affichage en format long, le premier caractère indique si c'est un fichier simple ('-') ou un dossier ('d'). Pourtant j'ai déjà vu d'autres symboles. C'était quoi ?

Il existe d'autres types de fichiers que les deux que nous avons vus. Ils se rencontrent moins souvent et sont surtout utilisés par le système. Par exemple, certains définissent des *pilotes* vers le matériel. Si vous voulez en savoir plus, vous pouvez lire ceci (en.wikipedia.org/wiki/Unix_file_types).

Vous avez mentionné les permissions 'r', 'w' et 'x'. Pourtant j'ai déjà vu d'autres lettres dans la zone réservée aux permissions. C'était quoi ?

Il y a 3 permissions dont nous n'avons pas parlé parce qu'elles sont moins courantes : le *suid* (set user id), le *sgid* (set group id) et le *sticky*. Si vous voulez en savoir plus, vous pouvez lire ceci (fr.wikipedia.org/wiki/Permissions_UNIX).

J'ai un Linux à la maison et les groupes ne sont pas les mêmes. C'est normal ?

Oui ! Les groupes dépendent à la fois de la distribution particulière utilisée et de la façon dont l'administrateur (le root) a configuré le système.

Vous n'avez pas expliqué le sens de la 2^e colonne fournie par la commande ls (juste avant le propriétaire) ?

C'est vrai mais c'est moins utile et plus lié à la structure interne du système de fichier. Je veux bien vous dire qu'il s'agit du nombre de liens physiques sur le fichier mais je sens que vous commencez déjà à regretter d'avoir posé la question ;)

Nous avons vu qu'un fichier est créé avec des permissions par défaut. C'est configurable ?

Oui. Voyez la commande `umask`.

3 Conclusion

Notions importantes de ce TD

Voici les notions importantes que vous devez avoir assimilées à la fin de ce TD.

- ▷ Pouvoir demander plus d'informations sur une commande, ses options. Comprendre les informations fournies.
- ▷ La notion de propriétaire de fichier et la notion de groupe.
- ▷ Pourvoir interpréter le résultat de la commande `ls -l`.
- ▷ Comprendre que les permissions sont données à 3 catégories de personnes et comprendre ce que ces catégories signifient.
- ▷ Comprendre les 3 sortes de permissions pour un fichier et pour un dossier.
- ▷ Pourvoir modifier le groupe et les permissions (avec des nombres et avec des lettres).

Exercice 13

Exercice récapitulatif

TODO...

Félicitations !

Vous êtes arrivé au bout de ce premier TD. Avant de quitter le laboratoire, n'oubliez pas de quitter proprement la connexion avec `linux1` (`exit`) et d'éteindre l'ordinateur (ou au moins de vous déloguer).

À la semaine prochaine et soyez à l'heure !