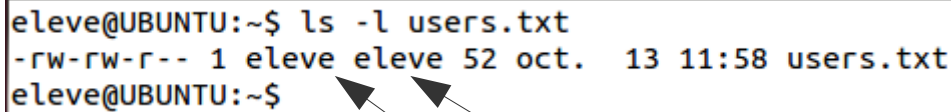


## Afficher les droits sur un fichier/répertoire

La commande **ls -l** affiche des informations complètes sur les fichiers/répertoires

```
eleve@UBUNTU:~$ ls -l users.txt
-rw-rw-r-- 1 eleve eleve 52 oct. 13 11:58 users.txt
eleve@UBUNTU:~$
```



Les principales informations retournées sont :

- le premier tiret indique que **users.txt** est un fichier ; on aurait un **d** s'il s'agissait d'un répertoire ;
- **r w - r w - r - -** indique les droits **UGO User Group Others** qui s'appliquent au fichier **users.txt** ;
- **User** correspond au **propriétaire** du fichier qui est **eleve** ici (3e valeur de la ligne) ;
- **Group** correspond au **groupe** qui possède le fichier, c'est le groupe **eleve** ici (4e valeur de la ligne) ;
- **Others** correspond à tous les autres utilisateurs (les autres c'est à dire ni le propriétaire ni un membre du groupe) ;
- les droits qui s'appliquent sont
  - r = droit de lecture = valeur 4 en décimal ;
  - w = droit d'écriture = valeur 2 en décimal ;
  - x = droit d'exécution = valeur 1 en décimal ;
- pour le fichier **users.txt**
  - l'utilisateur/propriétaire **U** (c'est à dire **eleve**) a le droit de lecture et d'écriture ;
  - les membres du groupe **G** (c'est à dire le groupe **eleve**) ont le droit de lecture et d'écriture ;
  - les autres **O** (qui ne sont pas l'utilisateur **eleve** et ne font pas partie du groupe **eleve**) ont le droit de lecture.

### Remarques :

- le droit **x** sur un répertoire permet de se déplacer dans ce répertoire et d'accéder aux fichiers sur lesquels on a des droits ;
- le droit **rx** sur un répertoire permet d'afficher son contenu ;
- le droit **wx** permet d'ajouter des fichiers dans le répertoire ;
- on peut envoyer le résultat d'une commande dans un fichier à l'aide des symboles **>** et **>>** qui sont des redirections :
  - **ls -l > ls.txt** crée le fichier **ls.txt** s'il n'existe pas, sinon le fichier **ls.txt** est écrasé ;
  - **echo "terminé" >> ls.txt** ajoute le texte "terminé" à la fin du fichier **ls.txt** s'il existe, sinon crée le fichier **ls.txt** qui contient "terminé".

## Modifier les droits sur un fichier/répertoire

Pour modifier les droits sur un fichier ou un répertoire, on utilise la commande **chmod**

### Exemples :

- **chmod u-w fic.txt** retire le droit d'écriture sur le fichier **fic.txt** à l'utilisateur propriétaire du fichier ;
- **chmod u+w fic.txt** ajoute le droit d'écriture sur le fichier **fic.txt** à l'utilisateur propriétaire du fichier ;
- **chmod u=w fic.txt** ne met que le droit d'écriture sur le fichier **fic.txt** à l'utilisateur propriétaire du fichier ;

Syntaxe : **chmod [ugo]{+|-|=}[rwx]**

Il existe une notation décimale pour attribuer les droits :

- r = 4
- w = 2
- x = 1

On peut ainsi combiner les droits en additionnant les valeurs décimales :

- un droit de lecture + écriture correspond à 6 ;
- un droit de lecture + exécution correspond à 5 ;
- un droit d'écriture seul correspond à 2 ;
- ...

Avec les droits en décimal, on modifie les droits UGO en ne tenant pas compte des droits initiaux.

#### Exemples :

- `chmod 777 fic.txt` donne tous les droits à tout le monde sur le fichier `fic.txt` ;
- `chmod 222 fic.txt` donne le droit d'écriture à tout le monde sur `fic.txt` ;
- `chmod 000 fic.txt` retire tous les droits à tout le monde.

**Remarque :** pour changer les droits sur un répertoire et tout ce qu'il contient en une seule commande, on utilise **chmod -R NomRepertoire**

**Exercice 1 :** on dispose du fichier `mysql.log` ayant les droits `rw-r--r--` . Donnez les commandes permettant de modifier les droits sur ce fichier et donnez les droits résultant sur ce fichier

1. ajout écriture au groupe ; `chmod g+w mysql.log` / `chmod 664 mysql.log`
2. aucun droit aux autres ; `chmod o-r mysql.log` / `chmod 660 mysql.log`
3. ajout lecture aux autres ; `chmod o+r mysql.log` / `chmod 664 mysql.log`
4. exécution seule au propriétaire ; `chmod u+x mysql.log` / `chmod 764 mysql.log`
5. ajout exécution aux autres ; `chmod o+x mysql.log` / `chmod 765 mysql.log`
6. aucun droit à tout le monde ; `chmod ugo-rwx mysql.log` / `chmod 000 mysql.log`
7. lecture seule à tout le monde ; `chmod ugo+w mysql.log` / `chmod 222 mysql.log`
8. retrait lecture aux autres ; `chmod o-r mysql.log` / `chmod 660 mysql.log`
9. ajout écriture au propriétaire ; `chmod u+r mysql.log` / `chmod 660 mysql.log`
10. tous les droits aux propriétaires et lecture/écriture au groupe et aux autres. `Chmod ugo+rw mysql.log`  
`chmod u+x mysql.log` / `Chmod 766 mysql.log`

**Exercice 2 :** on exécute les commandes suivantes sur un système Linux

```

root@debian:~# tail -4 /etc/passwd
u1:x:1001:100::/home/u1:/bin/bash
u2:x:1002:100::/home/u2:/bin/bash
u3:x:1003:1003::/home/u3:/bin/bash
u4:x:1004:1004::/home/u4:/bin/bash
root@debian:~# grep -w 100 /etc/group
users:x:100:
root@debian:~# grep u1 /etc/group
root@debian:~# grep u2 /etc/group
root@debian:~# grep u3 /etc/group
u3:x:1003:
root@debian:~# grep u4 /etc/group
u4:x:1004:
root@debian:~# ls -l /home/u1/
total 4
drwxr-x--x 2 u1 users 4096 26 janv. 20:36 REPU1
root@debian:~# ls -l /home/u1/REPU1/
total 0
-rw-r--r-- 1 u1 users 0 24 janv. 20:16 fic1
-rw-r--r-- 1 u3 u3 0 24 janv. 17:58 fic2
root@debian:~# █

```

- expliquez chacune des commandes ainsi que le résultat obtenu ;
  - 1) permet d'afficher les 4 premières lignes du fichier /etc/passwd
  - 2) s
  - 3) permet de chercher le mot u1 dans le dossier /etc/group
  - 4) permet de chercher le mot u2 dans le dossier /etc/group
  - 5) permet de chercher le mot u3 dans le dossier /etc/group
  - 6) permet de chercher le mot u4 dans le dossier /etc/group
  - 7) regarder le dossier /home/u1/
  - 8) regarder le dossier /home/u1/
- donnez la commande tapée par u1 pour ne donner que le droit d'écriture sur fic1 aux autres ; **chmod o=w fic1**
- écrire les droits sur le fichier fic1 ;
 

u peut lire et écrire  
g peut lire  
o peut lire
- u3 peut-il taper la commande cd/home/u1/REPU1 ?
 

Oui il peut se diriger vers ce dossier
- u3 peut-il taper la commande ls /home/u1/REPU1 ?
 

Oui il peut taper la commande pour voir l'intérieur du dossier
- u3 peut-il modifier le contenu de fic1 ? Si oui comment ?
 

Non il ne peut pas car il est en tant qu'autre et donc en lecture
- u4 peut-il modifier le contenu de fic2 ? Si oui comment ?
 

Non il ne peut pas car il est en autre et donc en lecture
- u3 peut-il lire le contenu de fic2 ? Si oui comment ?
 

Oui car il est le propriétaire du dossier et en faisant un « ls -l » on peut voir qu'il est en lecture et en écriture
- u1 peut-il autoriser u2 à ajouter des fichiers dans REPU1 ? Si oui comment ?
 

non
- u2 peut-il lire le fichier fic2 ?
 

Si oui comment ? Oui car les autres sont en lecture seulement

## Modifier le propriétaire et le groupe d'un fichier/répertoire

---

Pour changer le propriétaire d'un fichier/répertoire, on utilise la commande **chown (change owner)** :

**chown utilisateur fichier/repertoire**

Pour changer le groupe d'un fichier/répertoire, on utilise la commande **chgrp (change group)** :

**chgrp groupe fichier/repertoire**

On peut changer l'utilisateur ou le groupe d'un fichier/répertoire si on lance la commande en tant que **root** ou si on est le propriétaire du fichier/répertoire.

**Exercice 3** : recréez les utilisateurs, les groupes, les répertoires et les fichiers de l'exercice 2 en appliquant les mêmes droits, propriétaires et groupes aux fichiers/répertoires. Vous pouvez créer un utilisateur de la façon suivante :

**useradd -m -g users -s /bin/bash u1**

Ici on crée l'utilisateur **u1** qui aura un home directory (**option -m**), fera partie du groupe **users (option -g users)** et dont le shell de connexion sera **bash (option -s /bin/bash)**.