

[JOUR 4] I. Dompter son ordi : Permissions et utilisateurs. Cas pratique

Pratiquons un peu ce que je vous explique depuis le début !

1) Créons un utilisateur

Tout d'abord, créez un fichier dans un dossier personnel. Vous pouvez le faire avec la commande suivante :

```
echo "salut" > mon_fichier.txt
```

(Je détaillerai ce que fait cette commande plus tard.)

En exécutant `ls -l`, vous verrez que vous êtes le propriétaire du fichier et que vous disposez des permissions suivantes :

```
-rw-rw-r--
```

Mettons-nous dans un cas amusant où seul le propriétaire a les droits de lecture et d'écriture, et les autres utilisateurs n'ont aucun accès :

```
sudo chmod 600 mon_fichier.txt  
# Résultat après un ls -l : -rw-----
```

```
-rw----- 1 florent groupe_test  0 déc.  4 22:52 mon_fichier.txt  
florent@florent-Precision-5530:~/Documents/perso/data-crafting.io$
```

Créons maintenant un utilisateur nommé **test** :

```
sudo adduser test
```

Le programme vous guidera pour créer l'utilisateur. Choisissez un mot de passe simple de 8 caractères et notez-le si nécessaire. Pour les informations supplémentaires, appuyez simplement sur **Entrée** pour laisser les champs vides.

```
florent@florent-Precision-5530:~/Documents/perso/data-crafting.io$ sudo adduser test
Adding user `test' ...
Adding new group `test' (1002) ...
Adding new user `test' (1001) with group `test' ...
Creating home directory `/home/test' ...
Copying files from `/etc/skel' ...
New password:
```

```
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] Y
```

Pour vérifier que l'utilisateur a bien été créé, utilisez la commande suivante :

```
id test
# Résultat : uid=1001(test) gid=1002(test) groups=1002(test)
# Les identifiants peuvent varier chez vous.
```

Vous pouvez noter qu'un groupe avec le même nom que l'utilisateur a été créé. Certaines distros linux ont ce comportement pour améliorer la sécurité et la gestion des permissions en isolant les fichiers personnels de chaque utilisateur.

Par exemple cela simplifie la gestion des permissions, car les fichiers créés par l'utilisateur appartiennent par défaut à son groupe personnel.

Nous allons maintenant effectuer une substitution, mais pas pour prendre l'identité de l'utilisateur root avec `sudo`. Cette fois, nous allons devenir l'utilisateur **test** pour essayer de lire le fichier.

Pour cela, utilisez la commande `su` :

```
su test
```

Entrez le mot de passe de l'utilisateur **test**.

Une fois le mot de passe saisi, l'invite de commande devrait changer pour afficher "test" au début de la ligne.

```
su: Authentification refusée
florent@florent-Precision-5530:~/Documents/perso/data-crafting.io$ su test
Password:
test@florent-Precision-5530:/home/florent/Documents/perso/data-crafting.io$
```

Essayons maintenant de lire le fichier `mon_fichier.txt` avec la commande `cat` (qui affiche le contenu d'un fichier directement dans le terminal) :

```
cat mon_fichier.txt
```

```
Password:
test@florent-Precision-5530:/home/florent/Documents/perso/data-crafting.io$ cat mon_fichier.txt
cat: mon_fichier.txt: Permission denied
test@florent-Precision-5530:/home/florent/Documents/perso/data-crafting.io$
```

Vous constaterez que l'accès au fichier est refusé.

2) Créons un groupe

Qu'à cela tienne, créons un groupe avec les bonnes permissions et ajoutons l'utilisateur **test** à ce groupe.

Mais d'abord, revenons à notre utilisateur principal, car l'utilisateur **test** n'a pas les droits pour exécuter `sudo`.

```
su votre_user # Pour moi, c'est florent
# Tapez le mot de passe que vous utilisez pour déverrouiller
```

vosre session Linux.

Ok, maintenant créons un nouveau groupe :

```
sudo groupadd groupe_test
```

Vous pouvez vérifier que le groupe a bien été créé en consultant le fichier `/etc/group`. Pour rappel, **/etc** contient les fichiers de configuration du système et des programmes.

Affichez son contenu avec la commande `cat` :

```
cat /etc/group | grep groupe_test
# Résultat : groupe_test:x:1002:
# l'id du groupe peut varier chez vous
```

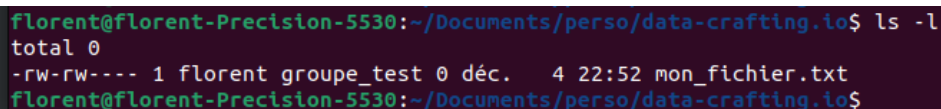
Super, notre groupe a été créé ! Changeons maintenant le groupe propriétaire du fichier :

```
sudo chown :groupe_test mon_fichier.txt
```

Faites un `ls -l` pour vérifier les modifications.

Ajoutons les droits de lecture et d'écriture pour le groupe :

```
sudo chmod g+rw mon_fichier.txt
```



```
florent@florent-Precision-5530:~/Documents/perso/data-crafting.io$ ls -l
total 0
-rw-rw---- 1 florent groupe_test 0 déc.  4 22:52 mon_fichier.txt
florent@florent-Precision-5530:~/Documents/perso/data-crafting.io$
```

À ce stade, le groupe dispose des permissions, mais il ne sert encore à rien car aucun utilisateur n'y a été ajouté.

Ajoutons l'utilisateur **test** au groupe **groupe_test** :

```
sudo usermod -aG groupe_test test
```

L'option `-a` signifie "append" (ajouter), et `-G` spécifie le groupe.

Pour vérifier que l'utilisateur a bien été ajouté au groupe, utilisez la commande suivante :

```
groups test
```

```
florent@florent-Precision-5530:~/Documents/perso/data-crafting.io$ sudo usermod -aG groupe_test test
florent@florent-Precision-5530:~/Documents/perso/data-crafting.io$ groups test
test : test groupe_test
florent@florent-Precision-5530:~/Documents/perso/data-crafting.io$
```

3) Testons l'accès au fichier

Faisons une dernière substitution avec l'utilisateur **test** :

```
su test
```

Essayons d'afficher le contenu du fichier :

```
cat mon_fichier.txt
# résultat : salut
```

```
test@florent-Precision-5530:/home/florent/Documents/perso/data-crafting.io$ cat mon_fichier.txt
salut
```

Et voilà, c'est gagné !