



UNIX

Cours 5 : Gestion des utilisateurs et droits d'accès

Filipe Vasconcelos¹

¹ESME, Lille, filipe.vasconcelo@esme.fr

- ① Gestion des utilisateurs**
- ② Gestion des droits**



AAV5

Analyser et explorer l'arborescence d'un système de fichiers d'un OS du type Unix à l'aide des commandes de base, et administrer les droits d'accès des répertoires et fichiers de l'arborescence.



① Gestion des utilisateurs

② Gestion des droits



- ❑ UNIX est un système d'exploitation multitâche et multi-utilisateur.
- ❑ Cela signifie que plusieurs personnes peuvent travailler simultanément sur le même OS, en s'y connectant à distance notamment.
- ❑ Il existe un ensemble de règles qui disent qui a le droit de faire quoi sur le système.
- ❑ Puisque plusieurs utilisateurs peuvent être connectés à Linux en même temps, il faut avoir un mécanisme pour organiser ces utilisateurs



- ❑ Chaque utilisateur doit s'identifier pour pourvoir accéder à la machine, il est désigné par un numéro unique qui s'appelle UID (User identifier)
- ❑ UID : identifiant (unique) de chaque compte utilisateur. Les nombres de 0 à 99 sont fréquemment réservés à des comptes propres à la machine. Les valeurs supérieures à 100 sont elles réservées aux comptes utilisateurs.
- ❑ Les utilisateurs sont regroupés dans des groupes identifiés par des numéros uniques qui s'appelle GID (Group Identifier).
- ❑ Les groupes servent à rassembler des utilisateurs afin de leur attribuer des droits communs.
- ❑ Chaque ressource (et donc fichier) appartient à :
 - ❑ Un seul utilisateur : celui qui a créé ce fichier donc il devient le propriétaire de ce fichier
 - ❑ Un ou plusieurs groupes d'autres utilisateurs



Type d'utilisateurs

Il existe deux types d'utilisateurs sur le système Unix :

- Utilisateur (normal) : celle qui va utiliser le système dans le but de produire quelque chose, le système UNIX est pour elle un moyen, un outil.
- Administrateur (super utilisateur) : Il est chargé de l'installation, de la configuration et de la bonne marche du système UNIX.

Sur un système UNIX, les utilisateurs UNIX ont des droits limités. C'est à dire que certaines commandes leurs sont interdites et ils n'ont pas accès à certaines parties du système. Les administrateurs systèmes ont par contre tous les droits sur le système.



- ❑ Sur tout système UNIX, il y a un super-utilisateur, généralement appelé root, qui a tous les pouvoirs sur le système.
- ❑ Il peut accéder librement à toutes les ressources de l'ordinateur, y compris à la place d'un autre utilisateur, c'est-à-dire sous son identité.
- ❑ En général, du moins sur les systèmes de production, seul l'administrateur système possède le mot de passe root. L'utilisateur root porte le UID 0.
- ❑ Cette simple protection permet de largement limiter les dégâts en cas de fausse manipulation, de virus sur votre PC, ...

sudo – « superuser do » ou « substitute user do »

- Cette commande permet d'exécuter une commande en tant que root ou un autre utilisateur
- Le mot de passe sera demandé pour exécuter la commande. Ce mot de passe est le même que celui de votre compte utilisateur limité.
- Exemple :

```
$ sudo apt-get update
```

```
[sudo] Mot de passe filipe :
```



- ❑ Cette commande permet de devenir root (superutilisateur) et le rester
- ❑ Le symbole # (nouveau prompt) à la fin de l'invite de commandes vous indique que vous êtes devenus superutilisateur.
- ❑ Pour quitter le « mode root », tapez exit



Commande adduser

- ❑ Cette commande permet d'ajouter un nouvel utilisateur.
- ❑ Il faut l'exécuter en tant que root (avec sudo ou rester en root)

```
$ sudo adduser toto
```

```
Ajout de l'utilisateur « toto » ...
Ajout du nouveau groupe « toto » (1003) ...
Ajout du nouvel utilisateur « toto » (1003) avec le groupe « toto » ...
Création du répertoire personnel « /home/toto »...
Copie des fichiers depuis « /etc/skel »...
Nouveau mot de passe :
BAD PASSWORD: The password is a palindrome
Retapez le nouveau mot de passe :
passwd: password updated successfully
Changing the user information for toto
Enter the new value, or press ENTER for the default
Full Name []: Le petit Toto
Room Number []: 42
Work Phone []: 555-4242
Home Phone []: 555-4242
Other []: Fait des blagues
Cette information est-elle correcte ? [0/n]0
```



- ❑ Le répertoire toto est crée sous /home
- ❑ Changer le mot de passe avec la commande : passwd

```
$ sudo passwd toto
```

Nouveau mot de passe :

BAD PASSWORD: The password is shorter than 8 characters

Retapez le nouveau mot de passe :

passwd: password updated successfully

- ❑ Chaque utilisateur appartient à un groupe
- ❑ Par défaut, si rien n'est spécifié, un groupe du même nom que l'utilisateur sera automatiquement créé.
- ❑ Exemple :

```
$ ls -l /home/
```

```
drwxr-xr-x 73 filipe filipe 4096 oct. 5 16:41 filipe
drwxr-xr-x  6 git git 4096 oct. 16 2019 git
drwxr-xr-x 17 guillaume guillaume 4096 janv. 2 2018 guillaume
```

Commande addgroup

- Cette commande permet de créer un nouveau groupe
- Exemple :

```
$ sudo addgroup unix
```

Ajout du groupe « unix » (GID 1003)...
Fait.

- Pour l'instant, il n'y a aucun utilisateur dans ce groupe



Commande usermod

- ❑ Cette commande permet d'éditer un utilisateur
- ❑ -g : changer de groupe
- ❑ Exemple : Pour mettre filipe dans le group unix :

```
$ sudo usermod -g unix filipe
```

Commande delgroup

Cette commande permet de supprimer un groupe



Commande deluser

- ❑ Cette commande permet de supprimer un utilisateur existant.
- ❑ **Ne supprimez surtout pas votre propre compte utilisateur !! Sinon, vous ne pourrez plus vous connecter lors de démarrage du système**
- ❑ Avec option `-remove-home`, le répertoire personnel de toto sous `/home` sera aussi supprimé

```
$ sudo deluser --remove-home toto
```

Recherche des fichiers à sauvegarder ou à supprimer...

Suppression des fichiers...

Suppression de l'utilisateur « toto »...

Attention ! Le groupe « toto » ne contient plus aucun membre.

Fait.

1 Gestion des utilisateurs

2 Gestion des droits



- ❑ Sous Unix les utilisateurs sont divisé en trois classes :
 - ❑ user (u) : le propriétaire du fichier
 - ❑ group (g) : le groupe (ou les groupes) associés au fichier
 - ❑ others (o) : le reste du monde
- ❑ Il est possible de définir des droits d'accès différents pour chacune de ces 3 catégories.
- ❑ L'administrateur du système (généralement root) a tous les droits pour manipuler les fichiers.



- Nom du fichier ou répertoire
- Date avec heure de la dernière de modification
- Taille en octets
- Groupe d'appartenance
- Propriétaire
- Le nombre de liens
- Droits d'accès (neuf caractères)
- Type de fichier (un caractère)



Droits d'accès d'un fichier

Exemple :

```
-rwxr-xr-x 1 filipe filipe 12 oct. 6 12:17 test_acces.sh
```

Droits du propriétaire :

- Lecture (r)
- Ecriture (w)
- Exécution (x)

rwx r-x r-x



Droits du groupe :

- Lecture
- Exécution

Droits des autres :

- Lecture
- Exécution



Droits d'accès d'un répertoire

Exemple :

```
drwxr-xr-x 2 filipe filipe 4096 oct. 6 16:55 rep_test_acces
```

rwx r-x r-x



Droits du propriétaire :

- Lecture (de lister le contenu) (r)
- Ecriture (de créer un nouveau fichier dans le répertoire) (w)
- Exécution (d'accéder avec un cd) (x)

Notez : Si on attribue w, il faut attribuer aussi x sur le répertoire



Commande de base : chown

- ❑ Signification : **change owner**
- ❑ change le propriétaire du fichier/repertoire
- ❑ Syntaxe : `chown [-R] nouveau_propriétaire [fichier|repertoire]`
- ❑ -R : modification récursives des sous-dossiers
- ❑ Exemple :

```
$ chown toto fic.txt
```



Commande de base : chgrp

- ❑ Signification : **change group**
- ❑ change le groupe du fichier/repertoire
- ❑ Syntaxe : chgrp [-R] nouveauPropriétaire [fichier|repertoire]
- ❑ -R : modification récursives des sous-dossiers
- ❑ Exemple :

```
$ chgrp unix fic.txt
```



Commande de base : chmod

- ❑ Signification : **change mode**
- ❑ Modification des droits des fichiers et répertoires
- ❑ Syntaxe : `chmod [-R] droits [fichier|repertoire]`
- ❑ `-R` : modification récursives des sous-dossiers
- ❑ `-v` : verbose
- ❑ Les droits peuvent être donnés sous forme symbolique ou numérique



chmod droits fichier

l'argument droits est une combinaison de trois symboles :

Utilisateur :

- u (user)
- g (group)
- o (other)
- a (all)

Modificateur :

- + (ajout)
- (supression)
- = (affectation)

Droite :

- r (read)
- w (write)
- x (execute)

Exemples :

- chmod a+x test.txt
- chmod go-rw test.txt
- chmod u=rw test.txt
- chmod u+x,go=rw test.txt



Droits	Binaire	décimal
rwx	111	7
rw-	110	6
r-x	101	5
r-	100	4
-wx	011	3
-w-	010	2
-x	001	1
—	000	0

Exemples :

chmod 755 test.txt

chmod u=rwx,g=r-x,o=r-x test.txt

