

Administration des utilisateurs



1 GESTION DES UTILISATEURS

Dans un système Linux, il existe toujours en plus des utilisateurs « **ordinaires** » le « **super-utilisateur** » qui s'appelle « **root** » et qui a tous les droits.

Les utilisateurs sont regroupés en groupes, auxquels on peut assigner des permissions. Seul **root** peut créer des utilisateurs, des groupes, et les modifier.

Chaque utilisateur est identifié par son « **login** », identifiant texte (32 car. ASCII max.) et son **UID** (User Identifier), entier unique. De façon similaire chaque groupe est identifié par une chaîne et par un identifiant numérique, le **GID** (Group Id). Un utilisateur peut appartenir à plusieurs groupes.

Liste des commandes permettant la gestion des utilisateurs :

- ✓ **useradd xxx** : création d'un utilisateur d'identifiant « **xxx** » ;
option -m pour créer en même temps le répertoire personnel de l'utilisateur
option -g ggg pour définir le groupe « **ggg** » comme groupe principal de l'utilisateur
- ✓ **groupadd xxx** : création d'un groupe d'identifiant « **xxx** »
- ✓ **passwd [xxx]** : changement de mot de passe de l'utilisateur « **xxx** »
- ✓ **usermod xxx** : modification des propriétés de l'utilisateur « **xxx** » (voir **man**)
option -g ggg pour modifier le groupe principal de l'utilisateur par le groupe « **ggg** »
option -G ggg pour ajouter l'utilisateur au groupe secondaire « **ggg** »
- ✓ **deluser xxx** : supprimer le compte « **xxx** »
- ✓ **delgroup xxx** : supprimer le groupe « **xxx** »
- ✓ **su [xxx]** : démarre un sous-shell avec le compte « **xxx** » (« **root** » si « **xxx** » est absent)
- ✓ **exit** : quitte le sous-shell en cours et retourne au shell parent
- ✓ **id [xxx]** : affiche l'**UID** et le **GID** de l'utilisateur « **xxx** »
- ✓ **groups [xxx]** : affiche les groupes auquel l'utilisateur « **xxx** » appartient
- ✓ **whoami** : affiche le nom de l'utilisateur courant
- ✓ **chown** : modifier le propriétaire d'un dossier ou d'un fichier

Pour l'ensemble de ces commandes, si l'argument optionnel est absent, alors c'est l'utilisateur courant qui sera concerné par la commande, sauf précision.

Les informations sur les utilisateurs sont centralisées dans le fichier **/etc/passwd**. Ce fichier contient une ligne par utilisateur, avec 7 champs séparés par ":" :

```
root:x:0:0:root:/root:/bin/bash
```

↓ ↓ ↓ ↓ ↓ ↓ ↓
1 2 3 4 5 6 7

1. **Username** : identifiant textuel
2. **Password** : un « **x** » indique que celui-ci est chiffré, dans le dossier **/etc/shadow**
3. **User ID** (UID)
4. **Group ID** (GID)
5. **User ID Info** : commentaire additionnel
6. **Home** de l'utilisateur

7. **shell** : chemin de l'interpréteur à utiliser pour cet utilisateur

Les informations sur les groupes sont stockées dans le fichier **/etc/group**

Exercice :

Pour cet exercice, vous devez être administrateur (root). Pour cela, vous pouvez utiliser une version linux gratuite dans le navigateur internet.

1. Créer deux groupes **group1** et **group2**
2. Créer 2 utilisateurs **glu1** et **glu2** du groupe **group1**. Le mot de passe de l'utilisateur 1 est **user1g1** et le mot de passe de l'utilisateur 2 est **user2g1**.
3. Créer 2 utilisateurs **g2u1** et **g2u2** du groupe **group2**. Le mot de passe de l'utilisateur 1 est **user1g2** et le mot de passe de l'utilisateur 2 est **user2g2**.
4. Vérifier à partir des fichiers **/etc/passwd** et **/etc/group** les groupes des utilisateurs **glu1**, **glu2**, **g2u1** et **g2u2**.

Manipulation de fichier :

- a. **glu1** : essayer de modifier le contenu des fichiers **/etc/passwd** et **/etc/group**
- b. **glu1** : Créer le fichier **File** dans le répertoire de connexion et modifier ses droits d'accès de sorte qu'ils seront égaux à **rw-r-----**
- c. **glu2** : Afficher le contenu du fichier **File** (créé par **glu1**) et essayer de le modifier
- d. **g2u1** : Refaire b.
- e. **glu2** : Copier le fichier **File** (de l'utilisateur **glu1**) dans son répertoire de connexion.
- f. **g2u1** : Copier le fichier **File** (de l'utilisateur **glu1**) dans son répertoire de connexion.
- g. **glu1** : Voir les droits d'accès du répertoire de connexion et interpréter.
- h. **glu2** : Essayer de supprimer le fichier **F** (qui se trouve dans le compte de l'utilisateur **glu1**) et créer un autre fichier **G** dans le compte de l'utilisateur **glu1**.

Manipulation de répertoires :

- a. **glu1** : Créer le répertoire **rep1** dans le répertoire de connexion et modifier ses droits d'accès de sorte qu'ils seront égaux à **rwxr-x---**
- b. **glu2** : Afficher le contenu du répertoire **rep1** (créé par **glu1**) et créer un fichier test. Interpréter le résultat obtenu.
- c. **g2u1** : Refaire b. Interpréter le résultat obtenu.
- d. **glu1** : Attribuer à son répertoire de connexion les droits suivants : **rwxrwxr-x**
- e. Refaire b et c. Interpréter.
- f. **glu1** : Changer le propriétaire du fichier **liste** à **g2u1**.

- g. **g1u1** : Changer le groupe du fichier *CopieListe* à **g2**
- h. Ajouter l'utilisateur **g2u1** au group 1 comme un group secondaire.
- i. **g1u1** : Créer un répertoire **rep1** et donner les droits aux membre de son groupe.
- j. **glu2 et g2u1** : vérifier l'accès au répertoire.
- k. **g1u1** souhaite créer un répertoire partagé que avec **g2u1**. Comment faire ?

2 CREATION D'ARCHIVE

La commande universelle pour gérer des archives dans l'**environnement Linux** est « **tar** ». Une archive est un fichier contenant un ensemble de fichiers avec leur chemin, ainsi que les métadonnées associées. On peut en plus avoir une compression non destructrice.

Cette commande est très utile, par exemple pour sauvegarder des données utilisateurs présents sur un serveur.

Syntaxe :

```
tar -[options] archive.tar [fichiers]
```

avec :

- ✓ **archive.tar** : nom du fichier archive
- ✓ **[fichiers]** est la liste des fichiers ou dossiers à archiver
- ✓ **[options]** étant une liste de commandes, parmi lesquelles
- ✓ **f** (obligatoire) : indique que le 1er argument est le nom du fichier
- ✓ **c** : création de l'archive
- ✓ **v** : ("Verbose") affiche la liste des fichiers au fur et à mesure
- ✓ **x** : extraire les fichiers
- ✓ **z** : ajout de la compression « **zip** »
- ✓ **j** : ajout de la compression « **bzip2** » (taux de compression supérieur)

Exemples :

- ✓ Création :

```
tar -cvf MesPhotos.tar /home/user1/Photos
```

- ✓ Extraction (dans le dossier courant) :

```
tar -xvf MesPhotos.tar
```

Questions :

- Archivez votre dossier scripts sous le nom backup.tar
- Archivez le même dossier sous format zip sous le nom backup.tar.gz
- Refaite la question précédente avec cette fois-ci une compression bzip2 sous le nom backup.tar.bz2. Pour ce type de compression, il faut installer au préalable le package bzip2.

dnf install bzip2.

- Comparez les tailles des trois archives

3 AUTRES COMMANDES

Alias

On peut créer des alias sur des commandes complexes, pour éviter les erreurs et aller un peu plus vite ...

1. Tapez la commande **alias**. Combien d'alias sont définis par défaut ?
2. Essayez la commande

```
alias toto='echo toto !'
```

Apparaît-elle dans la sortie de **alias** ?

3. Définissez l'alias **rc** comme un raccourci pour la commande

```
ls -al / wc -l,
```

testez votre alias.

4. Ouvrez une sous session à votre nom (avec su). Est-ce que vous pouvez utiliser votre alias ?

Pour que vos alias soient persistant, vous devez modifier le fichier **.bashrc** en ajoutant votre alias dans la section réservée à cet effet. Fermer votre session et ouvrez là de nouveau, est-ce que votre alias est toujours utilisable ?