**Exercice   
Permissions sur les objets du système de fichiers**

**Objectif**

Dans cet exercice, vous vous familiariserez avec les permissions et les droits Linux.

**Rappel** …  
On peut toujours obtenir de l’aide sur une commande à l’aide des modules d’aide.  
Comme exemple, pour obtenir de l’aide sur la commande **uname** …   
**>> uname --help**

**Remarque** …  
Pour les exercices suivants, l’utilisateur ayant ouvert une session est **tux**.  
L’utilisateur **tux** est également membre du groupe **sudo**.

**Étape 01**Lancez la commande **id** …  
**>> id**

On constate que l'utilisateur tux possède le **UID 1000** et le **GID** (groupe primaine) **1000**.

**Étape 02**Lancez les commandes suivantes pour créer un nouveau fichier et un nouveau répertoire …  
**>> touch mon-fichier**  
**>> mkdir mon-repertoire**

Affichez leur permission à l'aide de la commande **ls -ld**.   
Le commutateur **-d** (forme longue **--directory**) permet d'afficher les permissions des répertoires eux-mêmes et non leur contenu …  
**>> ls -ld /etc**

Portez attention aux permissions octroyées par défaut aux nouveaux fichiers et aux nouveaux répertoires.

* Pour un nouveau fichier, les permissions par défaut sont **rw-rw-r--** ;
* Pour un nouveau répertoire, les permissions par défaut sont **rwxrwxr-x** ;
* Par défaut, l'utilisateur propriétaire est l'utilisateur qui a créé le fichier ou le répertoire ;
* Par défaut, le groupe propriétaire est le groupe primaire (ou principal) de l'utilisateur qui a créé le fichier ou le répertoire.

**Étape 03**Rappelez-vous que la commande ls -l affiche toujours les 3 ensembles de permissions   
(propriétaire – *user*, groupe propriétaire et Autres – *others*).

Rappelez-vous également qu'il y a trois permissions de base sur un fichier ou un répertoire …

* **lecture** (r ou *read*) ;
* **écriture** (w ou *write*) ;
* **exécution** (x ou *execute*).

**Rappel** …  
Les permission de base ont une signification différente si elle concerne un fichier ou un répertoire …

|  |  |  |
| --- | --- | --- |
| Permission | Effet sur un fichier | Effet sur un répertoire |
| **Lecture** (r) | Permet la lecture ou l'affichage du contenu du fichier Le contenu peut être visualisé ou copié | Les noms de fichiers dans le répertoire peuvent être listés D'autres détails ne sont pas disponibles |
| **Écriture** (w) | Permet l’écriture du contenu de sorte que les modifications apportées à un fichier peuvent être enregistrées  Notez que l'autorisation **w** nécessite aussi l'autorisation **r** sur le fichier pour fonctionner correctement | Des fichiers peuvent être ajoutés ou supprimés dans le répertoire Notez que l'autorisation **w** nécessite l'autorisation **x** sur le répertoire pour fonctionner correctement |
| **Exécution** (x) | Permet à un fichier d'être exécuté (généralement utilisé pour un script ou un programme binaire) | Permet à l'utilisateur d'utiliser la commande cd pour se déplacer dans le répertoire ou d'utiliser le répertoire dans un chemin pour accéder aux fichiers et à ses sous-répertoires |

**Étape 04**Soit le fichier **/etc/passwd** …  
**>> ls -ld /etc  
-rw-r--r-- 1 root root 1345 jan 7 10:24 /etc/passwd**

Pour savoir quel ensemble de permission d'un fichier s'applique à l'utilisateur courant (celui avec lequel vous avez ouvert un interpréteur de commande (*shell*)), il faut se poser les questions suivantes …

L'utilisateur courant est-il le propriétaire du fichier ?  
Pour le déterminer, lancez la commande **whoami** ou observez l’invite de commande (*prompt*).   
Comparez ensuite cette valeur au nom du propriétaire retourné par la commande **ls -l**.  
**>> whoami**Pour ce fichier, l'utilisateur propriétaire est **root**, donc **tux** n'est pas propriétaire du fichier et l'ensemble de permission applicable n'est pas celui de l'utilisateur propriétaire.

L'utilisateur courant est-il membre du groupe propriétaire du fichier ?  
Pour le déterminer, lancez la commande **groups** ou **id**.   
Comparez ensuite les groupes dont l'utilisateur est membre au groupe propriétaire retourné par la   
commande **ls -l** **…**  
**>> groups  
tux sudo …**

La liste ne contient pas le groupe root donc **tux** n'est pas membre du groupe propriétaire du fichier et l'ensemble de permission applicable n'est pas celui du groupe propriétaire.

Alors quelles permissions s'appliquent ?  
Dans cet exemple, l'utilisateur **tux** n'est ni le propriétaire du fichier, ni membre du groupe propriétaire.   
Par conséquent, l'ensemble de permissions qui s'applique à cet utilisateur pour le fichier **/etc/passwd** est celui de tous les autres utilisateurs.

La seule permission de l'utilisateur **tux** sur le fichier est de le **lire** (r).  
En effet, il est possible de lire le contenu du fichier et de l'afficher à l'écran.   
Sa modification est toutefois interdite.

**Étape 05**Soit le fichier **/etc/shadow** dont les permissions sont les suivantes …  
**>> ls -l /etc/shadow  
-rw-r----- 1 root shadow 816 jan 7 10:25 /etc/shadow**

Encore une fois, l'utilisateur **tux**, n'est ni l'utilisateur propriétaire, ni membre du groupe **shadow**.   
Les permissions pour tous les autres utilisateurs s'appliquent.  
Ici, aucune autorisation n'est accordée à cet ensemble de permissions.  
Pour cette raison, l'utilisateur **tux** ne peut pas afficher le contenu du fichier à l'écran (ni l'afficher, ni l'exécuter).

**Étape 06**Pour afficher les permissions du répertoire personnel de l'utilisateur **tux** …  
**>> ls -ld /home/tux/  
drwxr-xr-x 11 tux tux 18 avr 13 20:26 /home/tux/**

On remarque que **tux** en est le propriétaire et qu'il dispose de tous les droits sur ce répertoire, soit de **lister**, **ajouter/supprimer** **des fichiers** et de **positionner son interpréteur de commande** (*shell*) à l'intérieur avec la   
commande **cd**.

Les membres du groupe tux ont quant à eux les droits de **lister** et de **positionner leur** **interpréteur de commande** à l'intérieur seulement, sans pouvoir y écrire.

**Étape 07**En tant que tux, supprimez le répertoire mon-répertoire …  
**>> rm -r mon-repertoire/**

Cette opération est possible grâce aux permissions du propriétaire

**Étape 08**Sous Linux, le propriétaire d'un fichier peut modifier les ensembles de permissions comme il le souhaite.   
Il peut ajouter ou supprimer des droits aux autres utilisateurs, aux membres du groupe propriétaire, voire à lui-même.

Pour démontrer cela, lancez la commande **chmod u-w** pour retirer à l'utilisateur propriétaire **tux** le droit d'écrire dans son répertoire (la permission **w**) …  
**>> chmod --verbose u-w /home/tux/**

**>> mkdir test  
mkdir: cannot create directory ‘test’: Permission denied**

**>> rm -r mon-fichier  
rm: cannot remove 'mon-fichier': Permission denied**

On remarque que toute modification à l'intérieur du répertoire **/home/tux** est refusée, même à son propriétaire.

En tant que propriétaire, l'utilisateur peut toutefois s'accorder à nouveau les droits d'écriture sur un fichier ou un répertoire …  
**>> chmod --verbose u+w /home/tux/  
>> ls -ld /home/tux/**

**>> rm -r mon-fichier**

Souvenez-vous que la permission de supprimer un fichier ou un répertoire dépend des permissions sur le répertoire qui le contient et non sur le fichier lui-même.

**Étape 09**Créez un utilisateur grangallo ...  
**>> sudo useradd --shell /bin/bash --create-home grangallo**

Donnez-lui ensuite un mot de passe …  
**>> sudo** **passwd grangallo  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully**

**Étape 10**Ajoutez un groupe nommé travail, puis ajoutez l'utilisateur tux et l'utilisateur grangallo dans ce groupe …  
**>> sudo groupadd travail  
>> sudo usermod --append --groups travail tux  
>> sudo usermod --append --groups travail grangallo**

Validez que le groupe est correctement créé et que les deux utilisateurs en font partie …  
**>> grep travail /etc/group**

On constate que le groupe travail a (sur ce système) l'ID **n** et que les deux utilisateurs en sont membres.

Il faut maintenant fermer la session actuelle et en ouvrir une nouvelle afin que l’ajout au groupe **travail** soit effectif.

**Étape 11**Ouvrez un interpréteur de commande avec votre utilisateur personnel.  
Utilisez la commande **su -** ...  
**>> su - grangallo**

Validez les informations concernant votre utilisateur et les groupes dont il est membre.  
**>> whoami  
>> groups**

La commande **id** peut également être utilisée pour retourner le même résultat.

**Étape 12**Accédez au répertoire /home comme suit et listez les permissions des répertoires personnels des utilisateurs   
du système …  
**>> cd /home/  
>> ls -l**

Remarquez le troisième ensemble de permissions **r-x** qui confère à tous les (autres) utilisateurs la permission de lister et de se positionner dans n'importe quel répertoire personnel.

**Étape 13**En tant qu'utilisateur **grangallo** les permissions qui s'appliquent sur son répertoire personnel sont les suivantes …  
**drwxr-xr-x 10 grangallo 16 avr 13 21:15 grangallo**

Il est effectivement possible d'accéder à ce répertoire en tant que **grangallo** et de lire les fichiers qu'il contient.   
L'écriture est quant à elle interdite car la permission **w** n'est pas accordée à cet ensemble …  
**>> cd grangallo  
>> ls -l**

**mkdir: cannot create directory ‘test’: Permission denied**

Quittez l’interpréteur de commande de **wally** pour revenir en tant que **tux** …  
**>> exit**

**Étape 14**Une solution pour empêcher les autres utilisateurs d'accéder à notre répertoire personnel consiste à retirer les permissions de lecture (r) et d'exécution (x) sur notre répertoire personnel.  
Autrement dit, il s'agit de retirer toutes les permissions à l'ensemble des autres utilisateurs.

Pour ce faire, l'utilisateur propriétaire d'un fichier ou d'un répertoire à toujours la possibilité de changer ses permissions. Pour ce faire, on utilise la commande la **chmod**.

Pour retirer les permissions r et x à l'ensemble des autres utilisateurs - *others* (o) …  
**>> chmod --verbose o-rx /home/tux/  
>> ls -ld o-rx /home/tux/**

Cette commande utilise la méthode symbolique de la commande **chmod** en formulant sous la forme d'une équation la modification à apporter aux ensembles de permissions.

**Étape 15**La commande **chmod** admet une méthode plus puissante que la méthode symbolique utilisée à l'étape précédente.   
Il s'agit de la méthode **octale** ou **numérique** qui permet de définir d'un coup, à l'aide de trois chiffres, les permissions qu'on souhaite donner à chacun des trois ensembles.

L'utilisation de la méthode **octale** nécessite que les autorisations pour les trois ensembles soient spécifiées.   
Il est basé sur le système de numération octale (en base 8) dans lequel chaque type d'autorisation se voit attribuer une valeur numérique …

|  |  |
| --- | --- |
| Valeur numérique | Permission |
| 4 | Lire |
| 2 | Écrire |
| 1 | Exécuter |
| 0 | Aucune permission |

En additionnant la combinaison de nombres de **0 à 7**, toute combinaison possible d'autorisations de lecture, d'écriture et d'exécution peut être spécifiée par un seul chiffre.

|  |  |
| --- | --- |
| Valeur octale | Ensemble de permissions |
| 7 | rwx |
| 6 | rw- |
| 5 | r-x |
| 4 | r-- |
| 3 | -wx |
| 2 | -w- |
| 1 | --x |
| 0 | --- |

De façon équivalente, la commande **chmod 750 /home/tux/** aurait pu être utilisée pour remplacer la commande de l'étape 14 …  
**>> chmod --verbose 750 /home/tux/  
>> ls -ld /home/tux/**

**Étape 16**Ouvrez à nouveau un interpréteur de commande pour l’utilisateur **wally**.   
Utilisez la commande s**u -** …  
**>> su - tux**

Tentez maintenant de lister le contenu du répertoire personnel de **tux** …  
**>> ls /home/tux/**  
**ls: cannot open directory '/home/tux/': Permission denied**

Le répertoire est maintenant inaccessible pour quiconque qui n'est pas son propriétaire ou membre du groupe propriétaire.

**Étape 17**Supposons que l'utilisateur **tux** et **wally** doivent travailler sur des fichiers communs (lire et écrire).   
Pour donner à ces utilisateurs un répertoire de travail commun, il faudra ajouter **tux** et **wally** dans un groupe et rendre ce groupe propriétaire du répertoire de travail.

Supposons que nous souhaitons créer le répertoire **/partage** (à la racine) …  
**>> mkdir /partage**  
**mkdir: cannot create directory ‘/partage’: Permission denied**

À la lumière du résultat obtenu, on comprend que les permissions de la racine ne permettent pas de créer de nouveau répertoire à cet endroit …  
**drwxr-xr-x 21 root root 21 jan 7 02:54 /**

En effet, seul l'utilisateur propriétaire **root** possède la permission de créer un répertoire à la **racine** (/).

**Étape 18**Ouvrez un interpréteur de commande en tant que **root** et créez le répertoire **/partage** …  
**>> sudo su -**  
**>> mkdir /partage**  
**>> ls -ld /partage/**  
**drwxr-xr-x 2 root root 2 avr 14 11:37 /partage/**

On remarque que **root** et le groupe **root** sont les propriétaires de ce répertoire.

**Étape 19**Pour modifier l'utilisateur propriétaire d'un fichier il faut utiliser la commande **chown**.  
Seul **root** peut modifier l'utilisateur propriétaire d'un fichier.

Définissez l'utilisateur tux comme propriétaire comme suit …  
**>> sudo chown --verbose tux /partage/**

On constate que le changement a été apporté.   
Quittez le shell ouvert en tant que **root** …  
**>> ls -ld /partage/  
drwxr-xr-x 2 tux root 2 avr 14 11:37 /partage/**

**>> exit**

**Étape 20**Pour compléter la création du répertoire partagé entre tux et wally, il faut modifier le groupe propriétaire pour un groupe dont les deux utilisateurs sont membres.  
Ce groupe a été créé à l'étape 10 (travail) …  
**>> grep travail /etc/group**

Modifiez maintenant le groupe propriétaire du répertoire **/partage** pour le groupe **travail** …  
**>> chgrp --verbose travail /partage/**  
**>> ls -ld /partage/**  
**drwxr-xr-x 2 tux travail 2 avr 14 11:37 /partage/**

Le changement peut être fait en tant que **tux** parce qu'il est maintenant le propriétaire du répertoire **/partage** et membre du groupe travail.

**Étape 21**Accordez toutes les permission **rwx** au propriétaire et au groupe propriétaire, mais aucun droit aux autres utilisateurs. Utilisez la méthode numérique de la commande **chmod** …  
**>> chmod --verbose 770 /partage/**  
**>> ls -ld /partage/**  
**drwxr-x--- 2 tux travail 2 avr 14 11:37 /partage/**

**Étape 22**Les utilisateurs tux et wally ont maintenant pleinement accès à leur répertoire de travail partagé …  
**>> touch /partage/test-tux  
>> ls -l /partage/**  
**-rw-rw-r-- 1 tux tux 0 avr 14 11:52 test-tux**

**>> touch /partage/test-wally  
>> ls -l /partage/**  
**-rw-rw-r-- 1 tux tux 0 avr 14 11:52 test-tux  
-rw-rw-r-- 1 wally wally 0 avr 14 11:53 test-wally**

**Questions de révision et d'exploration**

**Question 01**

Soit le fichier id\_rsa suivant …  
**>> ls -l .ssh/id\_rsa**  
**-rw------- 1 tux sshd 1679 nov 23 16:16 .ssh/id\_rsa**

* Quel utilisateur est propriétaire de ce fichier ?  
  **--------------**
* Quel groupe est propriétaire de ce fichier ?  
  **--------------**
* Quelles sont les permissions accordées aux membres du groupe sur ce fichier ?  
  **--------------**

**Question 02**Considérez les permissions suivantes …  
**-r--r----- 1 tux tux 0 avr 14 12:05 apt.log  
-rw-rw-r-- 1 tux tux 0 avr 14 12:05 centos  
drwxrwxr-x 2 tux travail 2 avr 14 12:05 Exemple  
-rw-r--r-- 1 tux tux 0 avr 14 12:05 test.tar**

* Quelle est la valeur numérique (octale) des permissions pour chacun des fichiers ?  
  apt.log  **--------------**centos  **--------------**Exemple  **--------------**test.tar  **--------------**

**Question 03**Considérez les permissions du fichier nul et du répertoire qui le contient …  
**>> ls -ld . nul**  
**drwxr-xr-x 3 tux tux 7 avr 14 12:11 .  
---------- 1 tux tux 0 avr 14 12:11 nul**

* L'utilisateur **tux** peut-il supprimer le fichier nul ?   
  Pourquoi ?  
   **-------------- --------------**
* L'utilisateur **tux** peut-il changer le propriétaire de ce fichier ?   
  Pourquoi ?  
   **-------------- --------------**
* L'utilisateur **tux** peut-il changer le groupe propriétaire de ce fichier ?   
  Si oui, à quelle condition ?  
  **-------------- --------------**
* L'utilisateur **tux** peut-il changer les permissions de ce fichier ?   
  Pourquoi ?  
  **-------------- --------------**

**Question 04**Dans votre répertoire personnel, créez un fichier vide nommé **vide.py** à l'aide de la commande **touch** …  
**>> cd  
>> touch vide.py**

* Utilisez la commande **chmod** en mode **symbolique** pour ajouter la permission d'exécuter au propriétaire du fichier puis supprimer les autorisations d'écriture et d'exécution pour tous les autres.   
  Utilisez une seule commande **chmod** pour faire ce changement.   
  Quelle commande avez-vous saisie ?  
  **--------------**
* Quelle commande **chmod** en mode **octal** (numérique) aurait produit le même effet ?  
  **--------------**

**Question 05**Qui peut lire les fichiers ou afficher le contenu des répertoires suivants …

* /var/log/auth.log  
  **--------------**
* /tmp  
  **--------------**
* /root  
  **--------------**

Avec la collaboration de **Miguel Grandmont Champagne**