POLYTECH DIJON

TRAVAUX DIRIGES

SHELL

Droits utilisateurs & Premiers scripts

OBJECTIFS

Manipuler les commandes Shell sous Linux afin de :

- Comprendre les notions d'utilisateurs, de groupes et de droits.
- Se familiariser avec la rédaction de scripts Shell.

UTILISATEURS, GROUPES ET DROITS

Préparation du terrain

➢ Créez trois utilisateurs à l'aide de la commande « adduser » :

Utilisateur	Mot de passe
user1	pwduser1
user2	pwduser2
user3	pwduser3

Vous aurez besoin des privilèges administrateur pour exécuter cette commande : « sudo »

> **Astuce**
>
> N'hésitez pas à utiliser le **man** pour avoir des informations sur une commande que vous ne connaissez pas.

➢ Créez un dossier « documents » dans le dossier personnel de chaque utilisateur. Que remarquez-vous ? Utilisez « sudo » pour parvenir à vos fins.

➢ Ajoutez un fichier texte dans chacun des dossiers documents, fichier qui contiendra un texte différent pour chaque utilisateur.

Changement d'utilisateur

- Connectez-vous en tant que « user1 » grâce à la commande « su ».
- Déplacez-vous dans le dossier « documents » de « user1 ».
- Modifiez le contenu du fichier texte. Que remarquez-vous ?
- Essayez avec la commande « sudo ». Que remarquez-vous ?

A l'origine de tout, les droits

- Listez, de manière détaillée, le contenu du dossier « documents » de « user1 ».
- Qui est le propriétaire du fichier ?
- Quel groupe peut manipuler le fichier ?
- Quels sont les droits du propriétaire sur le fichier ?
- Quels sont les droits du groupe sur le fichier ?
- Quels sont les droits des autres utilisateurs sur le fichier ?
- Parmi les trois catégories précédentes, à laquelle appartenez-vous ?

Changement de propriétaire

Lorsque vous exécutez une commande en la précédant du mot clé « sudo », le système considère que c'est l'utilisateur « root » qui effectue l'action. De fait, si vous créez un fichier avec la commande « sudo », ce nouveau fichier appartiendra par défaut à « root ».

Commençons par remettre un peu d'ordre dans tout ça :

- Reconnectez-vous en tant que « esicorp » (commande « exit » pour quitter le mode « su »).
- Modifiez les propriétaires des dossiers « documents » et de leur contenu pour chaque utilisateur (commande « chown »).
- Reconnectez-vous en tant que « user1 » et tentez à nouveau de modifier le fichier texte.

Changement de groupe

Bien, chaque utilisateur a retrouvé le contrôle de ses fichiers. Voyons à présent la notion de groupe.

Sous Linux, la création d'un utilisateur entraîne la création d'un groupe d'utilisateurs qui porte le même nom. Ainsi, il existe un utilisateur « root » et un groupe d'utilisateurs « root ». De même, lorsque vous avez créé l'utilisateur « user1 », le système a créé un groupe d'utilisateurs nommé « user1 ».

- De la même façon que vous avez modifiez les propriétaires des dossiers « documents » à la question précédente, modifiez les groupes d'appartenance de ces mêmes dossiers pour que :

- Le groupe « user1 » soit associé au dossier « documents » (et son contenu) de « user1 ».
- Idem pour les utilisateurs « user2 » et « user3 ».

Ok, les groupes sont en place, mais… à quoi servent ces fameux groupes ? Vous l'avez vu, trois niveaux de privilèges sont configurables pour chaque fichier/dossier : le propriétaire du fichier, le groupe associé et les autres utilisateurs. De cette façon, il est possible de définir :

- ce que le propriétaire peut faire de son fichier,
- ce que les utilisateurs appartenant au groupe associé peuvent faire,
- et enfin, ce que les autres utilisateurs (ni propriétaires, ni membres du groupe) peuvent faire.

Modification des droits pour le groupe

- Modifiez les droits du fichier texte de « user1 » de manière que le groupe associé possède les droits en écriture (commande « chmod »).
- Ajoutez l'utilisateur « user2 » au groupe « user1 » (commande « usermod »)
- Connectez-vous en tant que « user2 ».
- Vérifiez que vous pouvez modifier le fichier texte de « user1 ».
- Faites de même avec « user3 ».

Le vilain petit canard

Vous avez vu qu'il était possible de permettre à plusieurs utilisateurs d'avoir les mêmes droits sur un fichier dont ils ne sont pas propriétaires en les ajoutant au même groupe.

- Connectez-vous en tant que « user1 »
- Créez un nouveau fichier texte dans le dossier « documents » de « user1 »
- Comment faire en sorte que seuls « user1 » et « user2 » puissent modifier ce nouveau fichier ? Oui, c'est bien cela, « user3 » sera mis de côté, mais uniquement pour ce fichier.
- Procédez et testez.

Nettoyage

- Supprimez les trois utilisateurs ainsi que leur dossier personnel.

PREMIER SCRIPT

Enchaîner les commandes en agitant vos petits doigts sur le clavier, c'est bien pour donner le change. Mais n'oubliez jamais que le Petit Robert comme le Grand Larousse s'accordent en tout point sur la définition suivante :

```
Informaticien, informaticienne (nom) : voir fainéant
```

Et comme tout fainéant qui se respecte, il est hors de question de retaper continuellement les mêmes suites de commandes.

Au lieu de cela, vous allez créer des scripts qui regrouperont un ensemble de commandes à exécuter dans un ordre bien précis. Plus efficaces, les scripts vous permettront de briller en société.

Commençons par le commencement

- Créez un dossier « scripts » dans votre dossier personnel (compte « esirem »)
- Dans ce nouveau dossier, créer un fichier « info.sh » dans lequel vous écrirez les lignes suivantes :

```
#!/bin/bash
echo “Informations système”
```

Le première ligne détermine le Shell à utiliser pour exécuter le script. Ici, c'est le Bourne-Again Shell (bash) qui est invoqué.

La seconde ligne affiche simplement la chaîne de caractères « Informations système ».

Pour qu'un script puisse être exécuté, il est indispensable que l'utilisateur qui l'exécute dispose des droits d'exécution sur ce fichier.

- Ajouter les droits d'exécution pour le propriétaire.
- Exécutez le script.

On ne va pas s'arrêter là !

Quitte à faire un « Hello World », autant faire quelque chose d'utile.

- Complétez le script précédent de manière à afficher :
  - La date du jour
  - L'utilisateur qui exécute le script
  - Le dossier de travail actuel
  - Le nom de la machine
  - L'adresse IP de la machine

Vous devez obtenir un affichage comme celui-ci :

```
Informations système ----------------------
 * Date :  07/10/2018
 * Utilisateur :  user1
 * Répertoire actuel :  /home/user1/documents/scripts
 * Nom de l'ordinateur :  DESKTOP-4E07A75
 * Adresse IP :  127.0.1.1
-----------------------------------------
```

Astuce : vous pouvez concaténer une chaîne de caractères et le résultat d'une commande de la manière suivante :

```
echo “Texte : ” $(commande)
```

- Testez le script
- Testez le script avec « sudo ». Que remarquez-vous ?

Ma commande à moi !

Vous l'avez noté, pour exécuter votre script vous devez, au choix, saisir le chemin complet d'accès au fichier ou un chemin relatif commençant par « ./ ». Ça se fait, mais ne serait-il pas possible de procéder comme pour les commandes de base comme « ls » ou « cat » ?

Il est possible d'ajouter votre dossier « scripts » au PATH du système. Cette variable d'environnement contient tous les dossiers dans lequel le système va tenter de trouver les commandes que vous lui demandez. Le problème, c'est qu'en ajoutant un de vos dossier au PATH, il est possible de rendre accessible un script malicieux sur votre machine, simplement en le plaçant dans votre dossier « scripts ». Ce n'est donc pas top pour la sécurité.

Au lieu de cela, nous allons créer un lien vers votre script dans le dossier /user/bin. Ce dossier possède deux avantages :

- Il est déjà présent dans le PATH
- Il faut avoir des privilèges administrateur pour le modifier

- Renommez votre fichier « info.sh » en « info ».
- Exécutez la commande suivante :

```
ln -s /home/esirem/scripts/info /usr/bin/info
```

Ceci a pour objectif de créer un lien symbolique dans le dossier /usr/bin. Ce lien se comporte comme un alias de votre fichier info.

- Placez-vous à la racine de l'arborescence.
- Exécutez la commande « info ».
- Faites de même avec « sudo ».