

Outils de base

Objectifs d'apprentissage

1. Savoir utiliser les commandes de base de Linux : pwd, ls, cd, mv, mkdir, rm, sudo.
2. Savoir consulter les pages de manuel de Linux (commande man).
3. Savoir interpréter et modifier les droits d'un fichier.
4. Savoir afficher et gérer des processus.

Contenu

Ce premier laboratoire vous permet de vous familiariser avec les outils que nous allons utiliser par la suite.

1. Linux
2. Les pages du manuel Linux (man)

Rapport à fournir

Pour ce laboratoire, aucun rapport n'est demandé.

1 Introduction

L'objectif premier du cours Téléinformatique de base est de vous permettre de comprendre le fonctionnement des réseaux informatique. Que se passe-t-il quand on envoie des données à travers un réseau complexe ? Vous allez voir que beaucoup de mécanismes comme DHCP, DNS, le routage, les retransmissions, etc., sont nécessaire pour rendre ça possible.

Ce premier laboratoire et le laboratoire suivant vous permettent d'apprendre les outils que nous allons utiliser par la suite.

2 Matériel

Nous allons utiliser une configuration très simple : un PC Linux connecté à Internet.

3 Exercices

Une partie des laboratoires demande l'utilisation de Linux. Cette section vous permet d'apprendre les bases de l'utilisation de Linux.

3.1 Comptes d'utilisateurs

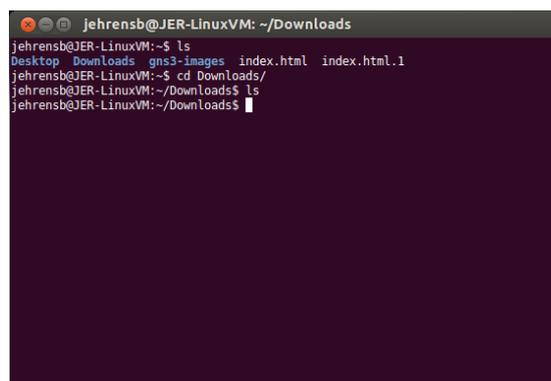
Le compte d'utilisateur normal sur les machines du laboratoire est :

- Login : labo
- Mot de passe : labo

L'administrateur d'un système Linux a le login « root ». Sur les PC du laboratoire, le mot de passe est aussi « root ».

Typiquement il n'est pas nécessaire de se logger directement comme root. Ne le faites pas !

Il est important de maîtriser la ligne de commande de Linux. Pour cela, ouvrez un terminal dans lequel vous pouvez taper les commandes.



```
jehrensb@JER-LinuxVM: ~/Downloads
jehrensb@JER-LinuxVM:~$ ls
Desktop Downloads gns3-images index.html index.html.1
jehrensb@JER-LinuxVM:~$ cd Downloads/
jehrensb@JER-LinuxVM:~/Downloads$ ls
jehrensb@JER-LinuxVM:~/Downloads$
```

3.2 Navigation dans les répertoires

Tout d'abord il est utile de savoir dans quel répertoire on se trouve. Vous pouvez utiliser la commande suivante :

```
pwd
```

Ensuite nous aimerions savoir ce qu'il y a dans le répertoire dans lequel nous nous trouvons :

```
ls -a
```

Le répertoire dans lequel vous vous trouvez est votre répertoire personnel (« home directory ») si vous avez un compte mais il s'agit d'un répertoire banalisé si vous avez accédé au système à l'aide d'un compte commun (comme dans notre cas avec « labo »). Remarque : faites attention à la casse. Le système est sensible aux majuscules/minuscules.

On peut changer de répertoire avec la commande `cd` :

```
cd /etc
```

nous fait accéder au répertoire `/etc`. De façon générique la commande s'écrit

```
cd <nom du répertoire>
```

La racine de l'arborescence des fichiers s'appelle « `/` ». La commande

```
cd /
```

vous met tout en haut de l'arborescence. Le slash « `/` » au début indique un chemin absolu, à partir de la racine. Donc

```
cd /etc
```

va dans le répertoire `etc` directement sous la racine. La commande

```
cd init
```

va dans le répertoire `init` qui doit se trouver sous le répertoire actuel.

Quand nous voulons remonter l'arborescence (par exemple aller de `/etc/init` à `/etc`), nous pouvons utiliser la commande

```
cd ..
```

Depuis n'importe quel emplacement il est possible de revenir dans son répertoire de base (« home directory ») avec la commande

```
cd
```

3.3 Exécuter des commandes comme utilisateur root

Normalement vous allez utiliser Linux comme utilisateur normal (utilisateur labo). Parfois il est nécessaire d'exécuter des commandes comme super-utilisateur root. Ceci est possible avec la commande

```
sudo commande ...
```

Si par exemple vous voulez afficher le contenu du répertoire home de root, Linux le refuse pour un utilisateur normal :

```
ls -a /root
ls: cannot open directory /root: Permission denied
```

Mais vous pouvez le faire avec sudo :

```
sudo ls -a /root
[sudo] password for labo:
.  ..  .bashrc.config  .profile  .pulse  .pulse-cookie
```

Utilisez la commande sudo avec prudence !

3.4 Commandes utiles

Voici une liste de commandes utiles. Essayez les commandes que vous ne connaissez pas encore.

cd <i>rep</i>	<i>Change directory</i> Se déplacer dans le répertoire <i>rep</i>
ls ls <i>rep</i> ls -l	<i>List</i> Lister le contenu du répertoire actuel. Lister le contenu du répertoire <i>rep</i> Lister un répertoire, affichage comme table (mieux lisible que <i>ls</i>)
cp <i>source dest</i>	<i>Copy</i> Copier le fichier source dans dest. La destination peut être un fichier ou un répertoire.
mv <i>source dest</i>	<i>Move</i> Renommer source en dest. Les deux peuvent être des fichiers ou des répertoires.
mkdir <i>rep</i>	<i>Make directory</i> Créer un nouveau répertoire <i>rep</i>
rm <i>fichier</i> rm -r <i>rep</i>	<i>Remove</i> Effacer un fichier Effacer un répertoire de manière récursive (-r)

3.5 Pages de manuel Linux

Linux dispose de pages de manuel pour toutes les commandes.

Vous pouvez consulter les « pages man » (*'man'* de manuel) avec la commande suivante :

```
man ls
```

Et vous verrez apparaître une description de la commande *ls* avec tous les arguments et options possibles. Vous pouvez presser sur <return> (pas à pas) ou <espace> (page par page) pour faire avancer. Si vous voulez quitter le fichier en cours de route, pressez simplement « q » (quit).

La commande *man* est certainement la commande la plus utile. C'est une bonne habitude d'utiliser cette commande dès qu'on hésite sur une commande.

3.6 Questions

Pour tester vos connaissances de Linux voici quelques questions :

Quelles commandes faut-il utiliser pour

- Aller dans le répertoire */etc/network*
- Afficher le contenu de ce répertoire
- Aller dans le répertoire au-dessus
- Retourner dans votre répertoire home */home/labo*
- Créer un répertoire « *testdir* » dans votre répertoire
- Renommer *testdir* en *nouveaudir*
- Effacer le répertoire *nouveaudir*

4 Commandes Linux avancées

4.1 Afficher le contenu d'un fichier

Souvent il est utile d'afficher le contenu d'un fichier, par exemple un fichier de configuration, sans ouvrir un éditeur.

Comment faire ? Il y a plusieurs possibilités : vous pouvez utiliser *less* avec la syntaxe suivante :

```
less /etc/passwd
```

Vous pouvez presser sur <return> (pas à pas) ou <espace> (page par page) pour faire avancer. Si vous voulez quitter le fichier en cours de route, pressez simplement « q » (quit).

Une autre possibilité est la commande *cat*, qui envoie le texte d'un fichier sur la sortie standard :

```
cat /etc/passwd
```

4.2 Gestion des droits

Un aspect très important dans Linux est la manière dont on peut protéger l'accès aux fichiers et aux répertoires. L'administrateur a tous les droits sur tous les fichiers. Si on effectue la commande *ls -l*, nous obtenons par exemple :

```
ls -l /
total 67
drwxr-x--x 2 root root 2048 Aug 14 20:01 bin
drwxr-xr-x 3 root root 1024 Oct 8 10:15 boot
.....
```

Le « *root root* » dans les lignes signifie que les fichiers appartiennent à l'utilisateur *root* qui fait partie du groupe *root*.

Que signifient les lettres de la première colonne (*drwxr...*)? En fait il s'agit des droits d'accès aux fichiers. On a trois groupes. En allant de la gauche vers la droite, on a successivement

- les accès pour le propriétaire,
- les membres du groupe
- et les autres utilisateurs.

Le fait que tous les caractères soient appondus n'arrange pas les choses mais on peut diviser la première ligne de la façon suivante : `rw- r-x`. Dans ces trois groupes, on a trois lettres chaque fois. La première lettre `r` signifie « lecture », la deuxième (`w`) signifie « écriture » et la troisième (`x`) signifie « exécution ». Donc si on reprend l'exemple ci-dessus, on voit que « boot » est accessible en lecture, écriture et exécution pour le propriétaire (`root`), accessible en lecture et exécution pour les membres du groupe et accessible en exécution seulement pour les autres utilisateurs.

Il est possible de changer ces accès avec la commande `chmod`. On peut ajouter des droits (« + »), en retirer (« - ») ou en remplacer (« = ») pour le propriétaire (« u »), le groupe (« g »), les autres (« o »), ou tout le monde (« a »).

On peut aussi donner une valeur en octal pour l'attribution des accès :

`rw=7, rw-=6, r-x=5, r--=4, -wx=3, -w-=2, --x=1, ---=0.`

Questions

Que signifie la commande suivante ?

`> chmod a+rw boot`

Que signifie la commande suivante ?

`> chmod 760 boot`

On peut aussi faire changer le propriétaire d'un fichier avec la commande `chown`. La syntaxe est la suivante :

```
chown <nouveau propriétaire> <fichier>
```

4.3 Gestion de processus

Sur Linux, chaque processus a un identificateur numérique, le PID (*process identifier*). Ce numéro est important car il identifie de façon unique le processus. Nous pouvons faire la liste des processus actifs avec la commande `ps` (voir les « man pages » pour les options).

Questions

Comment afficher tous les processus du système ?

Comment afficher ceux qui appartiennent à l'utilisateur labo ?

Commande top

Une autre commande très pratique est la commande *top*. Elle affiche en continu les processus actifs, la charge de la machine, la mémoire occupée, etc. Vous pouvez sortir en tapant « q » (quit).

Commande kill

Nous pouvons terminer des processus avec la commande *kill*. Il faut indiquer à la commande le PID du processus à supprimer. La syntaxe est la suivante :

```
> kill -9 <pid>
```

Comme exercice,

1. Lancez Firefox
2. Dans un terminal, lancez la commande *top*
3. Lors de l'utilisation de Firefox, il apparaît dans la liste *top*
4. Quittez *top* avec la touche 'q'
5. Affichez tous vos processus avec la commande *ps -U labo*
6. Relevez le pid du processus firefox
7. Terminez Firefox depuis le terminal avec la commande *kill <pid>*.

Faites toutefois attention quand vous supprimez des processus car vous pouvez faire des dégâts car le processus que vous supprimez ne va pas pouvoir terminer sa tâche ! On peut utiliser ceci quand on écrit un programme comportant une boucle infinie par exemple.

Une commande par fois plus commode est la commande *killall* avec comme argument le nom du programme et non pas le pid. Il supprime tous les processus de ce nom. Par exemple :

```
killall firefox
```

termine *toutes* les applications firefox.

5 Exercices avancés

Les exercices de cette section ne sont pas obligatoires. Ils ne seront pas considérés pour la note.

Mais si vous avez de l'intérêt et du temps, je vous encourage à les faire, afin d'approfondir vos connaissances et apprendre encore d'avantage sur les technologies des réseaux.

5.1 Chaînage de commandes

Toutes ces commandes (tâches) sont appelées des processus avec Unix. On peut également connecter la sortie d'un processus à l'entrée d'un autre. Cette connexion se fait avec le symbole « | » (pipe, AltGr-1 ou AltGr-7). Par exemple :

```
cat /etc/passwd | less
```

On peut ainsi « chaîner » des commandes.

5.2 Recherche de fichiers

Une autre commande très pratique est la commande « grep » qui permet de chercher des bouts de chaînes de caractère dans des fichiers texte. On peut chercher le mot « list » dans notre fichier man_ls.txt :

```
cat /etc/passwd | grep labo
```

On peut aussi rechercher un fichier dans une arborescence (pas toujours évident à faire manuellement, surtout quand l'arborescence comporte de nombreux répertoires). On donne la racine de l'arborescence dans laquelle on veut chercher et une expression désignant le fichier qu'on désire rechercher. Si on désire rechercher notre fichier depuis la racine du système (tout en haut de l'arborescence), il faut faire la commande suivante :

```
find /etc -name passwd -print
```

Cette commande trouve tous les fichiers avec le nom passwd sous le répertoire /etc ou un sous-répertoire.

Approfondir vos connaissances de Linux

Si vous souhaitez vous plonger plus profondément dans Linux, le site « Learn Linux The Hard Way » (<http://nixsrv.com/llthw>) offre de nombreux exercices pratiques qui vous permettront de devenir un expert Linux.