

Le système Linux.

Introduction

Linux est un système d'exploitation :

- multi-utilisateur = plusieurs utilisateurs peuvent ouvrir une session simultanément ;
- multi-tâche = un utilisateur peut lancer plusieurs programmes en même temps ;
- ouvert = s'installe sur des plates-formes matérielles différentes ;
- modulaire = un utilisateur peut créer ses propres programmes en C et les intégrer au système ;
- paramétrable grâce à des commandes = il possède un langage de commandes offrant de nombreuses fonctionnalités.

Le prompt

Quand on lance le Terminal, on obtient une ligne de la forme

```
root@UBUNTU:~#
```

ou

```
eleve@UBUNTU:~$
```

Ces 2 lignes s'appellent **le prompt**. On ne peut pas le supprimer avec la touche Retour arrière.

Le prompt nous donne plusieurs informations :

- **root ou eleve** indique quel utilisateur est connecté ; pour le vérifier on peut taper la commande **echo \$USER**
- **ubuntu** indique le nom de l'ordinateur ; on peut le vérifier avec la commande **hostname**
- **~** indique l'endroit où on se trouve dans l'arborescence des fichiers ; ici **~** (tilde = Alt Gr + 2) désigne le **home directory** de l'utilisateur connecté, c'est à dire son répertoire personnel ; pour le vérifier on tape **pwd** qui nous donne le répertoire courant ;
- **#** indique que l'utilisateur est "administrateur" du système ;
- **\$** indique que l'utilisateur est un utilisateur "normal".

L'arborescence des fichiers

Sous Linux, la racine des répertoires et fichiers du disque dur est **/** : c'est l'équivalent du **C:** sous Windows. Voici un exemple des répertoires présents à la racine :

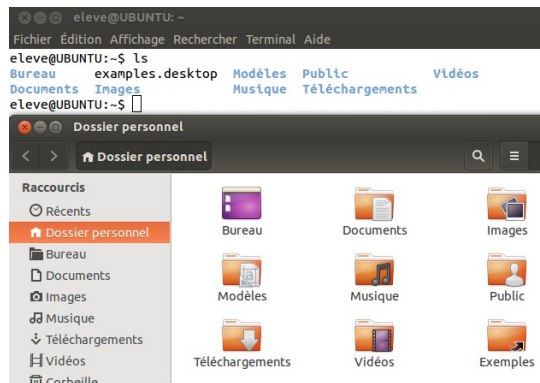
```
eleve@UBUNTU:~$ ls /
bin      dev      initrd.img      lib64      mnt      root      srv      usr      vmlinuz.old
boot     etc      initrd.img.old  lost+found  opt      run      sys      var
cdrom    home     lib             media      proc     sbin     tmp      vmlinuz
eleve@UBUNTU:~$
```

Le répertoire **/etc** contient les principaux fichiers de configuration du système.

Les commandes

Les actions que l'on effectue à l'aide de l'interface graphique sont pour la plupart réalisables à l'aide du **Terminal**. C'est un logiciel qui interprète les commandes saisies, c'est à dire les exécute et affiche le résultat à l'écran (comportement par défaut).

L'exemple le plus simple est d'afficher le contenu d'un répertoire, par exemple le répertoire personnel de l'utilisateur connecté :



La commande saisie est la commande **ls**. Son comportement par défaut est d'afficher le contenu du répertoire courant (répertoire dans lequel je me trouve).

On peut passer des options à la commande pour changer son comportement par défaut. Par exemple, on souhaite avoir plus d'informations sur les fichiers et les répertoires du répertoire courant.

```
eleve@UBUNTU:~$ ls -l
total 44
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Bureau
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Documents
-rw-r--r-- 1 eleve eleve 8980 sept.  3 22:06 examples.desktop
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Images
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Modèles
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Musique
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Public
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Téléchargements
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Vidéos
eleve@UBUNTU:~$
```

L'option **-l** permet de connaître le propriétaire du fichier/répertoire, sa taille, ...

On peut aussi passer des arguments à la commande, en plus d'une option. SI je veux connaître les fichiers/répertoires présents dans le répertoire `/etc/network`, avec les détails, on tapera la commande suivante :

```
eleve@UBUNTU:~$ ls -l /etc/network
total 24
drwxr-xr-x 2 root root 4096 sept.  3 22:25 if-down.d
drwxr-xr-x 2 root root 4096 sept.  3 22:21 if-post-down.d
drwxr-xr-x 2 root root 4096 sept.  3 22:20 if-pre-up.d
drwxr-xr-x 2 root root 4096 sept.  3 22:26 if-up.d
-rw-r--r-- 1 root root  82 sept.  3 22:02 interfaces
drwxr-xr-x 2 root root 4096 avril  4 2014 interfaces.d
lrwxrwxrwx 1 root root  12 sept.  3 22:03 run -> /run/network
eleve@UBUNTU:~$
```

`/etc/network` est un argument de la commande **ls**. La commande complète est donc **ls -l /etc/network**

Les utilisateurs

On peut créer des utilisateurs sous Linux à l'aide de la commande **adduser nomUtilisateur** Les utilisateurs ainsi créés possèdent un répertoire de travail personnel situé généralement dans le répertoire **/home**

Les informations sur les utilisateurs présents sur le système sont enregistrés dans le fichier **/etc/passwd**

Quand on crée un utilisateur, un groupe du même nom est créé. Exemple avec la commande **adduser afaret** :

- l'utilisateur **afaret** est créé ;
- le groupe **afaret** est créé avec pour seul membre l'utilisateur **afaret**.

Lien entre les utilisateurs et la commande ls

ls -l affiche des informations supplémentaires, comme on l'a vu précédemment.

```

eleve@UBUNTU:~$ ls -l
total 44
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Bureau
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Documents
-rw-r--r-- 1 eleve eleve 8980 sept.  3 22:06 examples.desktop
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Images
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Modèles
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Musique
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Public
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Téléchargements
drwxr-xr-x 2 eleve eleve 4096 sept.  3 22:11 Vidéos
eleve@UBUNTU:~$ █

```

Je crée un fichier avec la commande suivante : **echo "Prof. SI1 M. Dudek" > sio1b.txt**

```

eleve@UBUNTU:~$ ls -l sio1b.txt
-rw-rw-r-- 1 eleve eleve 19 sept. 29 10:42 sio1b.txt
eleve@UBUNTU:~$

```

Les principales informations retournées sont :

- le premier tiret indique que **sio1b.txt** est un fichier ; on aurait un **d** s'il s'agissait d'un répertoire ;
- **rw-rw-r--** indique les droits **UGO User Group Others** qui s'appliquent au fichier sio1b.txt ; les droits qui s'appliquent sont
 - **r** = droit de lecture = valeur 4 en décimal ;
 - **w** = droit d'écriture = valeur 2 en décimal ;
 - **x** = droit d'exécution = valeur 1 en décimal ;
- ici l'utilisateur/propriétaire **U** a le droit de lecture et d'écriture, le groupe **G** a le droit de lecture et d'écriture les autres **O** ont le droit de lecture ;
- le premier **eleve** indique que le fichier appartient à l'utilisateur **eleve** ;
- le second **eleve** indique que le fichier appartient au groupe **eleve**.

Pour modifier les droits sur un fichier ou un répertoire, on utilise la commande **chmod**

Exemples :

chmod u-w sio1b.txt retire le droit d'écriture sur le fichier sio1b.txt à l'utilisateur propriétaire du fichier

(syntaxe **chmod [ugo]{+|-}[rwx]**)

chmod 777 sio1b.txt donne tous les droits à tout le monde sur le fichier **sio1b.txt**

Les redirections

Le résultat d'une commande s'affiche par défaut à l'écran. On peut envoyer le résultat d'une commande (le rediriger) vers un fichier texte à l'aide du symbole **>** ou **>>**.

Voici la différence entre les 2 symboles :

```

eleve@UBUNTU:~$ ls
Bureau      exemples.desktop  Modèles  Public  Téléchargements
Documents  Images           Musique  sio1b.txt  Vidéos
eleve@UBUNTU:~$ ls > resultat.txt
eleve@UBUNTU:~$ ls
Bureau      exemples.desktop  Modèles  Public  sio1b.txt  Vidéos
Documents  Images           Musique  resultat.txt  Téléchargements
eleve@UBUNTU:~$ cat resultat.txt
Bureau
Documents
exemples.desktop
Images
Modèles
Musique
Public
resultat.txt
sio1b.txt
Téléchargements
Vidéos
eleve@UBUNTU:~$ █

```

En utilisant une simple redirection, **on crée** le fichier **resultat.txt** ici qui contient le résultat de la commande **ls**. Que se passe-t-il si le fichier **resultat.txt** existe déjà ?

```

eleve@UBUNTU:~$ cat resultat.txt
Bureau
Documents
exemples.desktop
Images
Modèles
Musique
Public
resultat.txt
sio1b.txt
Téléchargements
Vidéos
eleve@UBUNTU:~$ ls /var/log/apt > resultat.txt
eleve@UBUNTU:~$ cat resultat.txt
history.log
term.log
eleve@UBUNTU:~$ █

```

On constate ici que le fichier **resultat.txt** est écrasé, son contenu étant remplacé par le résultat de la commande **ls /var/log/apt**

Applications : expliquer chacune des commandes des 2 captures d'écran précédentes.

Et si on utilise la redirection >> ?

```

eleve@UBUNTU:~$ ls > resultat.txt
eleve@UBUNTU:~$ cat resultat.txt
Bureau
Documents
examples.desktop
Images
Modèles
Musique
Public
resultat.txt
sio1b.txt
Téléchargements
Vidéos
eleve@UBUNTU:~$ ls /var/log/apt >> resultat.txt
eleve@UBUNTU:~$ cat resultat.txt
Bureau
Documents
examples.desktop
Images
Modèles
Musique
Public
resultat.txt
sio1b.txt
Téléchargements
Vidéos
history.log

```

La première redirection > crée ou écrase le fichier **resultat.txt** tandis que la deuxième redirection >> ajoute le résultat à la fin du fichier : on parle de **concaténation**.

La redirection fonctionne avec toutes les commandes, notamment avec la commande **echo** qui affiche un message à l'écran. On peut ainsi créer un fichier avec le contenu souhaité :

```

eleve@UBUNTU:~$ echo "abcdefghij" > alphabet.txt
eleve@UBUNTU:~$ echo "lmnopqrstu" >> alphabet.txt
eleve@UBUNTU:~$ echo "vwxyz" >> alphabet.txt
eleve@UBUNTU:~$ cat alphabet.txt
abcdefghij
lmnopqrstu
vwxyz
eleve@UBUNTU:~$

```

Résumé :

- la redirection > **resultat.txt**
 - crée le fichier **resultat.txt** s'il n'existe pas,
 - écrase le fichier **resultat.txt** s'il existe.
- la redirection >>
 - crée également le fichier **fichier.txt** s'il n'existe pas (**à tester**),
 - ajoute le résultat de la commande à la fin du fichier **resultat.txt** s'il existe.

Les tubes

On parle également de **pipe**.

Les commandes de manipulation de fichiers travaillent avec des fichiers comme arguments. Par exemple **cat /etc/network/interfaces** le fichier est **/etc/network/interfaces** et on l'appelle argument de la commande **cat**

cat /etc/passwd affiche le contenu du fichier contenant les informations des comptes utilisateurs.

Pour savoir si le compte **eleve** a été créé sur le système, on exécute la commande **grep eleve /etc/passwd**

Il est possible de combiner (=enchaîner, faire suivre) deux (ou plusieurs) commandes à l'aide d'un **tube (pipe = symbole | = Alt Gr + 6)**. Pour vérifier si **eleve** est un compte présent sur le système on pourra écrire

```
cat /etc/passwd | grep eleve
```

Explications :

- la commande de gauche, le cat, ne change pas de comportement, on lui passe un argument qui est le nom du fichier ;
- la commande de droite, le grep, voit son comportement modifié car elle ne travaille plus qu'avec un seul argument, l'expression recherchée ;
- la commande **grep eleve** utilisée seule échoue ;
- le 2e argument attendu par la commande grep, le nom d'un fichier, a été remplacé par le résultat retourné par la commande à gauche du pipe, ici le résultat retourné par cat ;
- la commande **cat /etc/passwd | grep eleve** est donc ici équivalente à **grep eleve /etc/passwd**

Ce mécanisme est souvent utilisé :

- pour rechercher si un service est en cours d'exécution, par exemple Apache **ps aux | grep bash**
- pour savoir si un fichier est dans un répertoire **ls /etc/network | grep interfaces**