

# TP :L'interface système

## Objectifs :

- **Notion d'arborescence de fichiers.**
- **Chemin relatif – chemin absolu**
- **Commande de gestion des fichiers et répertoires**
- **Commandes de gestion des droits sur les fichiers et répertoire**

## utiliser la ligne de commande-le shell

On va apprendre à utiliser l'émulateur de terminal dit aussi console. Avant les interfaces graphiques et l'exécution de nombreuses tâches à l'aide de la souris ou même de son doigt, dans les années 70 le seul accès à un ordinateur était par le terminal. Savoir l'utiliser n'est pas un retour en arrière nostalgique. Il permet de mieux comprendre comment fonctionne l'ordinateur et avec un peu d'habitude, d'être plus rapide pour certaines tâches qu'avec la souris dans un environnement graphique. Le shell (coquille en anglais) est ce qui entoure le noyau du système d'exploitation. Les commandes qui vont être présentées sont celles utilisées sur un système Linux/Unix ou sur MacOS. Depuis Windows10, il est possible d'installer un terminal avec un noyau Linux permettant de s'en servir. Le shell que nous allons utiliser est le **Bash** (acronyme de *Bourne-Again shell* )

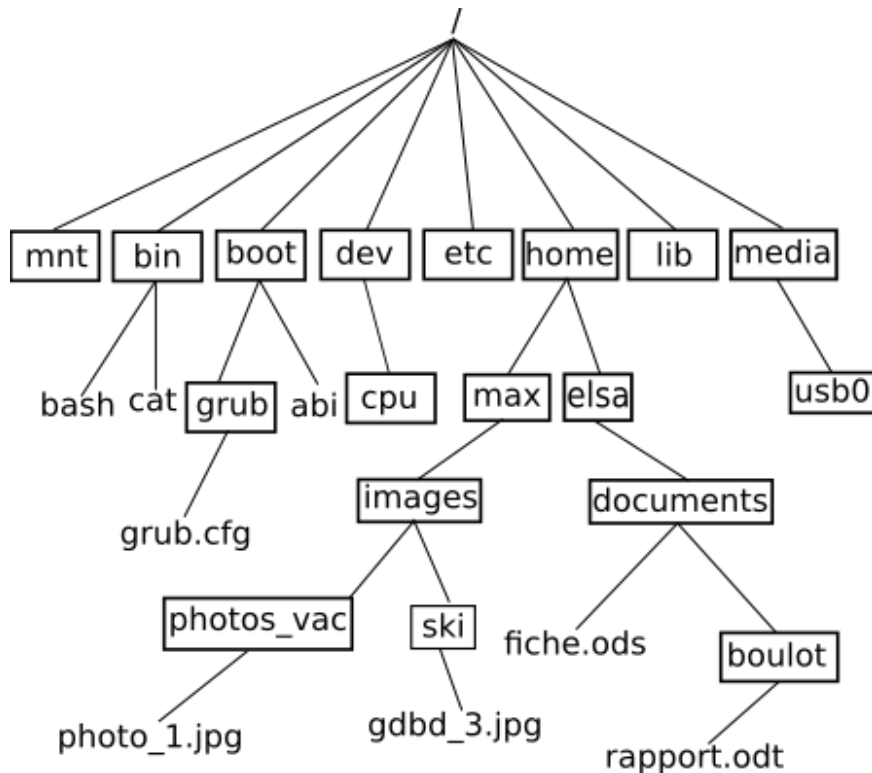
Nous allons voir que nous pouvons facilement agir sur les fichiers et les répertoires et les droits d'accès à ceux-ci. Il est possible de faire plein d'autres choses comme gérer les processus, faire des recherches dans des fichiers, aller sur Internet, gérer les droits des utilisateurs, etc.

## A Système de fichiers

Quand on parle de système de fichiers, on peut soit penser à l'organisation physique des données sur un support comme un disque dur, soit à la façon dont cette organisation va être représentée pour l'utilisateur. Ici, on ne s'intéressera qu'à la représentation pour l'utilisateur.

Dans la plupart des système d'exploitation, les fichiers sont regroupés dans des répertoires qui peuvent eux-mêmes contenir des répertoires. On dit que l'organisation des fichiers ce fait en arborescence. On parle d'arborescence de fichiers,car ce schéma ressemble à un arbre à l'envers. Dans un système « Unix », le répertoire qui contient tous les fichiers et les répertoires est appelé la racine et est noté « / » (« slash » en anglais). Le nom des fichiers et répertoire doit être sans espace que l'on remplace par un souligné « \_ » et en utilisant que les caractères ASCII.

Voici une arborescence possible de fichiers sous UNIX.



Dans le schéma ci-dessus, les répertoires comme « home » sont encadrés et les fichiers comme « bash » ou « photo\_1.jpg » ne le sont pas. On peut remarquer que dans les systèmes de fichiers de type « UNIX » les extensions de fichiers comme « .jpg » ne sont pas obligatoires. « cat » est une commande et un fichier. Dans un système « UNIX » « tout est fichier ».

## Chemin absolu – chemin relatif.

Pour indiquer la position d'un fichier ou d'un répertoire dans l'arborescence il y a deux méthodes : indiquer le chemin absolu, ou le chemin relatif.

### **Chemin absolu**

Le **chemin absolu** est le chemin en partant de la racine. Il commence donc par « / ».

Par exemple, le chemin absolu pour le fichier « fiche.ods » est : /home/elsa/documents/fiche.ods

Remarque : hormis le premier « / » qui indique la racine, les autres servent de séparateur.

### **Chemin relatif**

On peut aussi indiquer le chemin à partir d'un répertoire quelconque que l'on va appeler le répertoire de travail ou répertoire courant, on parle alors de **chemin relatif**. Le répertoire de travail est noté « . ». Le chemin est décrit à partir de ce répertoire. On peut ne pas écrire le « . » pour alléger l'écriture.

Par exemple, le chemin relatif pour atteindre « fiche.ods » depuis le répertoire « elsa » est : ./document/fiche.ods ou plus simplement document/fiche.ods .

Le chemin relatif pour atteindre répertoire « photos\_vac » depuis le répertoire « max » est :  
./images/photos\_vac ou images/photos\_vac.

On remarque qu'un chemin relatif ne commence pas par « / ».

Comment atteindre le répertoire « images » depuis le répertoire « elsa » ?

Pour cela nous devons « remonter d'un niveau dans l'arborescence », on dit aussi atteindre le répertoire parent, ce qui se fait à l'aide de deux points « .. ».

Ainsi pour atteindre le répertoire « images » depuis le répertoire « elsa », le chemin relatif est :  
../max/images.

Pour atteindre le répertoire « home » depuis le répertoire « elsa » le chemin relatif est : ../../

### Exercice 1

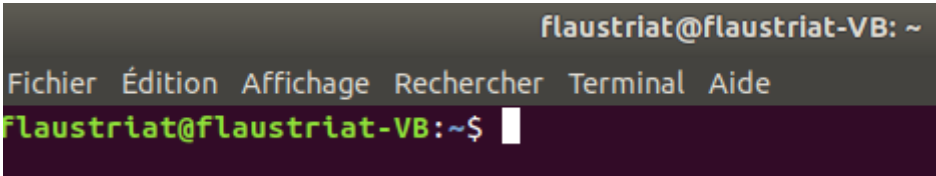
Donner le chemin absolu pour atteindre :

- le fichier « grub.cfg »
- le répertoire « boulot »

Donner le chemin relatif pour atteindre :

- Le fichier « rapport.odt » depuis le répertoire « documents »
- Le répertoire « boulot » depuis le répertoire « max »

Lorsque vous ouvrez une console, appelé aussi émulateur de terminal. Vous devez obtenir quelque chose comme :



```
flaustriat@flaustriat-VB: ~
Fichier Édition Affichage Rechercher Terminal Aide
flaustriat@flaustriat-VB:~$
```

Cette ligne signifie que l'utilisateur flaustriat est connecté à la machine flaustriat-VB. Le « ~ » indique que l'on est dans le répertoire de travail de cet utilisateur, ce qui correspond au chemin absolu /home/flaustriat. Le symbole « \$ » indique que l'on utilise le shell Bash.

Un système « UNIX » est multi-utilisateur. Chaque utilisateur a un répertoire à son nom qui contient ses propres fichiers et répertoires. Par convention, ce répertoire est dans le répertoire « /home ». Lorsqu'un utilisateur ouvre un terminal, celui-ci est dans le répertoire à son nom. Dans la copie d'écran nous avons : « flaustriat@flaustriat-VB ~\$ ».

Le caractère tilde « ~ » signifie que l'on est dans le répertoire de l'utilisateur courant, ici flaustriat. C'est le répertoire dont le chemin absolu est /home/flaustriat . Le répertoire où l'on se trouve actuellement est appelé le « répertoire courant » ou « répertoire de travail » (*working directory* en anglais).

L'invite de commande (*prompt* en anglais) vous indique en permanence le répertoire courant.

Par exemple « flaustriat@flaustriat-VB:~/Documents\$ » signifie que vous êtes dans le répertoire « Documents » du répertoire de l'utilisateur flaustriat. Son chemin absolu est donc /home/flaustriat/Document/s

Attention les systèmes « UNIX » sont « sensibles à la casse », il faut bien différencier majuscules et minuscules. Le répertoire « Documents » n'est pas le répertoire « documents ».

## Afficher le répertoire de travail – la commande pwd

La commande **pwd** (*Print Working Directory* en anglais) affiche le chemin absolu du répertoire courant.

## Changer de répertoire -la commande cd

La commande **cd** (*Change Directory* en anglais) permet de changer de répertoire courant. Il suffit de lui indiquer le chemin absolu ou relatif vers le nouveau répertoire courant.

Par exemple en utilisant l'arborescence vue plus haut :

- Si « max » est le répertoire courant pour que « images » soit le nouveau répertoire courant, il faut saisir : `cd images` en relatif ou `cd /home/max/images` avec le chemin absolu.
- Si « max » est le répertoire courant pour que « documents » soit le nouveau répertoire courant, il faut saisir ; `cd ../elsa/documents` en relatif ou `cd /home/elsa/documents`

## Exercice 2

Quelle commande devez-vous saisir si le répertoire courant est « home » et que vous vouliez que le nouveau répertoire courant soit « images » ? Vous donnerez la commande avec le chemin absolu, puis avec le chemin relatif.

## Liste les fichiers et répertoires - la commande ls

La commande **ls** (*LiSt* en anglais) permet d'afficher le contenu du répertoire courant.

```
flaustriat@flaustriat-VB:~$ cd Documents/
flaustriat@flaustriat-VB:~/Documents$ ls
3-Fichiers.pdf  NSI1
```

Le répertoire courant est celui de l'utilisateur flaustriat, la commande « cd Documents » change le répertoire courant en « /home/flaustriat/Documents ». « ls » liste le contenu du répertoire « Documents » il contient un fichier nommé « 3- Fichiers.docx » et un répertoire « NSI1 »

Taper ls chemin\_répertoire permet d'afficher le contenu du répertoire d'arriver du chemin qui peut être relatif ou absolue.

### Exercice 3

Dans une console taper la commande ls, puis ls Documents

### Créer un répertoire - la commande mkdir

La commande **mkdir** (*MaKe DIRectory* an anglais) permet de créer un répertoire nommé dans le répertoire courant. Pour créer un répertoire *nom\_répertoire*, on saisi : `mkdir nom_répertoire`

*Remarque* : Mieux vaut ne pas utiliser de caractères accentués dans les noms de répertoires et de fichiers. Il en est de même pour les espaces qu'il est préférable de remplacer par un « \_ ».

### Exercice 4

Dans le répertoire courant créer un répertoire nommé « test-NSI ».

### Effacer un fichier ou un répertoire -la commande rm

La commande **rm** (ReMove ) permet de supprimer un fichier ou un répertoire.

La commande est de la forme `rm nom_fichier`.

```
flaustriat@flaustriat-VB:~/Documents/test-NSI$ ls
fichier1.txt fichier2.txt
flaustriat@flaustriat-VB:~/Documents/test-NSI$ rm fichier1.txt
flaustriat@flaustriat-VB:~/Documents/test-NSI$ ls
fichier2.txt
flaustriat@flaustriat-VB:~/Documents/test-NSI$ cd ..
flaustriat@flaustriat-VB:~/Documents$ rm test-NSI/
rm: impossible de supprimer 'test-NSI/': est un dossier
```

Par défaut, la commande `rm` ne peut pas supprimer un répertoire. Comme beaucoup de commandes UNIX, `rm` a des options. Pour supprimer un répertoire non vide il faut ajouter l'option « -r ».

`rm -r nom_répertoire`

```
flaustriat@flaustriat-VB:~/Documents$ rm -r test-NSI/
flaustriat@flaustriat-VB:~/Documents$ ls
3-Fichiers.pdf NSI1
```

### Créer un fichier vide -la commande touch

La commande **touch** permet de créer un fichier vide. Pour créer un fichier nommé *nom\_fichier*

dans le répertoire courant, il faut saisir : touch *nom\_fichier*.

## Copier un fichier -la commande cp

La commande **cp** (*CoPy* en anglais) permet de copier un fichier. Elle s'utilise sous la forme :

cp *chemin\_vers/fichier1* *chemin\_répertoire\_arrive/fichier2*.

Le nom du nouveau fichier peut être différent de celui de départ.

```
flaustriat@flaustriat-VB:~/Documents$ mkdir test-NSI
flaustriat@flaustriat-VB:~/Documents$ cd test-NSI/
flaustriat@flaustriat-VB:~/Documents/test-NSI$ mkdir test
flaustriat@flaustriat-VB:~/Documents/test-NSI$ touch fichier1.txt
flaustriat@flaustriat-VB:~/Documents/test-NSI$ ls
fichier1.txt  test
flaustriat@flaustriat-VB:~/Documents/test-NSI$ cp fichier1.txt test/fichier2.txt
flaustriat@flaustriat-VB:~/Documents/test-NSI$ cd test/
flaustriat@flaustriat-VB:~/Documents/test-NSI/test$ ls
fichier2.txt
```

## Exercice 5

Dans le répertoire courant créer un répertoire « test-NSI ». Dans ce répertoire créer un fichier « *essai.txt* » et un répertoire « *tmp* ». Faire une copie du fichier « *essai.txt* » dans le répertoire « *tmp* » nommée « *essai2.txt* ». Retourner dans votre répertoire personnel puis effacer le répertoire « test-NSI » et son contenu.

## Déplacer un fichier -la commande mv

La commande **mv** (*MoVe* en anglais) permet de déplacer ou renommer un fichier.

```
flaustriat@flaustriat-VB:~/Documents/test-NSI/test$ ls
fichier2.txt
flaustriat@flaustriat-VB:~/Documents/test-NSI/test$ mv fichier2.txt fichier3.txt
flaustriat@flaustriat-VB:~/Documents/test-NSI/test$ ls
fichier3.txt
```

Le fichier « *fichier2.txt* » a été renommé « *fichier3.txt* ».

En cas de doute sur l'utilisation d'une commande vous pouvez-toujours utiliser la commande **man** (*MANuel* en anglais).

## Exercice 6

À l'aide du manuel de la commande **mv**, donner la commande qui permet de déplacer les fichiers « *fichiers1.txt* » et « *fichiers2.txt* » dans le répertoire « *test* » .

```
flaustriat@flaustriat-VB:~/Documents/test-NSI$ ls  
fichier1.txt fichier2.txt test
```

## Les répertoires fondamentaux dans un système Unix

Voici une liste de répertoires présent à la racine de l'arborescence et leur rôle dans un système UNIX

/

└bin : Commande de base du système (ls, cd, etc.).

└dev : Fichiers représentant les dispositifs matériels (devices) du système.

└etc : Fichiers de configuration systèmes

└home : Répertoire d'accueil des utilisateurs

└mnt : Point de montage (clé USB, DVD, etc.)

└root : répertoire de l'administrateur du système (séparé de celui des autres utilisateurs).

└usr : Logiciels installés

## B les droits sur les fichiers et répertoires

Les systèmes UNIX sont multi-tâches et multi-utilisateurs. Chaque utilisateur a donc des droits qui lui permettent de faire certaines actions et lui en interdisent d'autres. Le système d'exploitation permet de gérer ces droits finement.

Un utilisateur particulier nommé « super-utilisateur », « root » ou encore « administrateur » peut ajouter ou supprimer des droits à chacun des utilisateurs. Pour modifier les droits d'un ensemble d'utilisateurs, le super-utilisateur peut créer des groupes et modifier les droits pour l'ensemble d'un groupe.

Ici nous nous intéresserons qu'aux droits sur les fichiers, mais il existe des droits sur l'utilisation de l'imprimante, de la mémoire de stockage ou de la mémoire vive, de l'utilisation du processeur, etc.

Pour les fichiers et les répertoires, il y a trois type de droits.

- **r** (*Read* en anglais) symbolise le droit en lecture ;
- **w** (*Write* en anglais) symbolise le droit en écriture ;
- **x** (*eXecute* en anglais) symbole le droit d'exécution pour un programme ou un script.

UNIX sépare le monde en trois type d'utilisateurs :

- **u** (*user*) : L'utilisateur ;
- **g** (*group*) :Le groupe ;
- **o** (*others*) : Les autres .

Pour avoir des informations sur les droits des fichiers, il faut utiliser la commande « ls -l ».

```
flaustriat@flaustriat-VB:~/Documents/test-NSI$ ls -l
total 4
-rw-r--r-- 1 flaustriat flaustriat  0 févr. 14 16:00 fichier1.txt
-rw-r--r-- 1 flaustriat flaustriat  0 févr. 14 16:31 fichier2.txt
drwxr-xr-x 2 flaustriat flaustriat 4096 févr. 14 16:10 test
```

Étudions la première ligne :

« -rw-r--r-- 1 flaustriat flaustriat 0 févr. 14 16:00 fichier1.txt »

-rw-r--r-- se décompose ainsi en suivant l'ordre de lecture :

- le premier « - » indique que c'est un fichier, sinon il y aurait un « d » comme *directory*.
- rw- : l'utilisateur a les droits en lecture et écriture ;
- r-- : le groupe a le droit en lecture ;
- r-- : les autres ont le droit en lecture.

Le premier « flaustriat » est le nom de l'utilisateur, le second est celui de son groupe.

Étudions maintenant la dernière ligne :

« drwxr-xr-x 2 flaustriat flaustriat 4096 févr. 14 16:10 test »

- d : c'est un répertoire ;
- rwx : l'utilisateur a les droits de lecture, écriture et exécution ce qui veut dire pour un répertoire le droit de le traverser ;
- r-x: le groupe a les droits de lecture et d'exécution ;
- r-x : les autres ont les droits de lecture et d'exécution.

### Exercice 7

Analyser les droits sur le fichier3.txt.

« -rwxr--r-- 1 flaustriat flaustriat 0 févr. 14 17:43 fichier3.txt



## Modifier les droits d'un fichier -la commande chmod

La commande **chmod** (Change MODE en anglais) permet de changer les droits des fichiers et répertoires qui appartiennent à notre utilisateur.

Nous avons vu plus haut que nous allons utiliser les lettres, u,g et o pour le type d'utilisateurs et les lettres r, w et x pour le type de droits. Pour ajouter un droit on utilisera « + », pour le retirer on utilisera « - » et « = » pour revenir aux droits par défaut.

La syntaxe générale est :

```

                                chmod [u g o a] [+ - =] [r w x] nom_du_fichier
flaustriat@flaustriat-VB:~/Documents/test-NSI/test$ ls -l
total 0
-rw-r--r-- 1 flaustriat flaustriat 0 févr. 14 17:53 fichier.txt
flaustriat@flaustriat-VB:~/Documents/test-NSI/test$ chmod u-w+x fichier.txt
flaustriat@flaustriat-VB:~/Documents/test-NSI/test$ ls -l
total 0
-r-xr--r-- 1 flaustriat flaustriat 0 févr. 14 17:53 fichier.txt

```

Initialement l'utilisateur avec les droits de lecture et écriture sur fichier.txt, à la fin il a les droits en lecture et exécution sur ce fichier.

### Exercice 8 :

Depuis votre répertoire d'utilisateur créer un répertoire « test-NSI2 ». Créer dans celui-ci un fichier « test.txt » et un répertoire « test ». Dans le répertoire test créer un fichier « test2.txt ».

Placez-vous dans le répertoire « test-NSI2 » et supprimer les droits d'exécution du répertoire « test » pour l'utilisateur. Vérifier que le droit a été enlevé à l'aide de la commande « ls -l »

Tenter de vous placer dans le répertoire « test ». Remettre le droit que vous avez enlevé et tenter à nouveau de vous placer dans ce répertoire.

**Sources :** [https://pixees.fr/informatiquelycee/n\\_site/nsi\\_prem\\_cmd\\_base\\_linux.html](https://pixees.fr/informatiquelycee/n_site/nsi_prem_cmd_base_linux.html)

## Quelques commandes bash

Commande	Action
<code>pwd</code>	Affiche le chemin du répertoire courant
<code>ls</code>	Affiche les fichiers et dossiers du répertoire courant
<code>ls chemin_repertoire</code>	Affiche les fichiers et dossiers du chemin_repertoire
<code>ls -l</code>	Affiche les droits des fichiers et dossiers du répertoire courant
<code>cd chemin_repertoire</code>	Le répertoire courant devient chemin_repertoire
<code>mkdir nom_nouveau_repertoire</code>	Créer un répertoire nom_nouveau_repertoire dans le répertoire courant
<code>Touch nom_fichier</code>	Créer un fichier nom_fichier vide
<code>cp fichier1 fichier2</code>	Créer une copie du fichier fichier1 nommée fichier2
<code>mv fichier1 fichier2</code>	Déplace le fichier fichier1 en le renommant fichier2
<code>rm nom_fichier</code>	Efface le fichier nom_fichier
<code>rmdir nom_repertoire</code>	Efface le répertoire nom_repertoire uniquement s'il est vide.
<code>rm -r nom_repertoire</code>	Efface le répertoire nom_repertoire et son contenu

## Changements de droits sur un un fichier ou un répertoires

Syntaxe :

```
chmod [u g o a] [+ - =] [r w x] nom_du_fichier_ou_repertoire.
```

Utilisateurs concernés	Type de droits
u : utilisateur g : groupe o : les autres a : pour tous	r : lecture w : écriture x : exécution

Exemple

`chmod u+w-x fichier1` : ajoute le droit de lecture et retire le droit d'exécution du fichier fichier1