

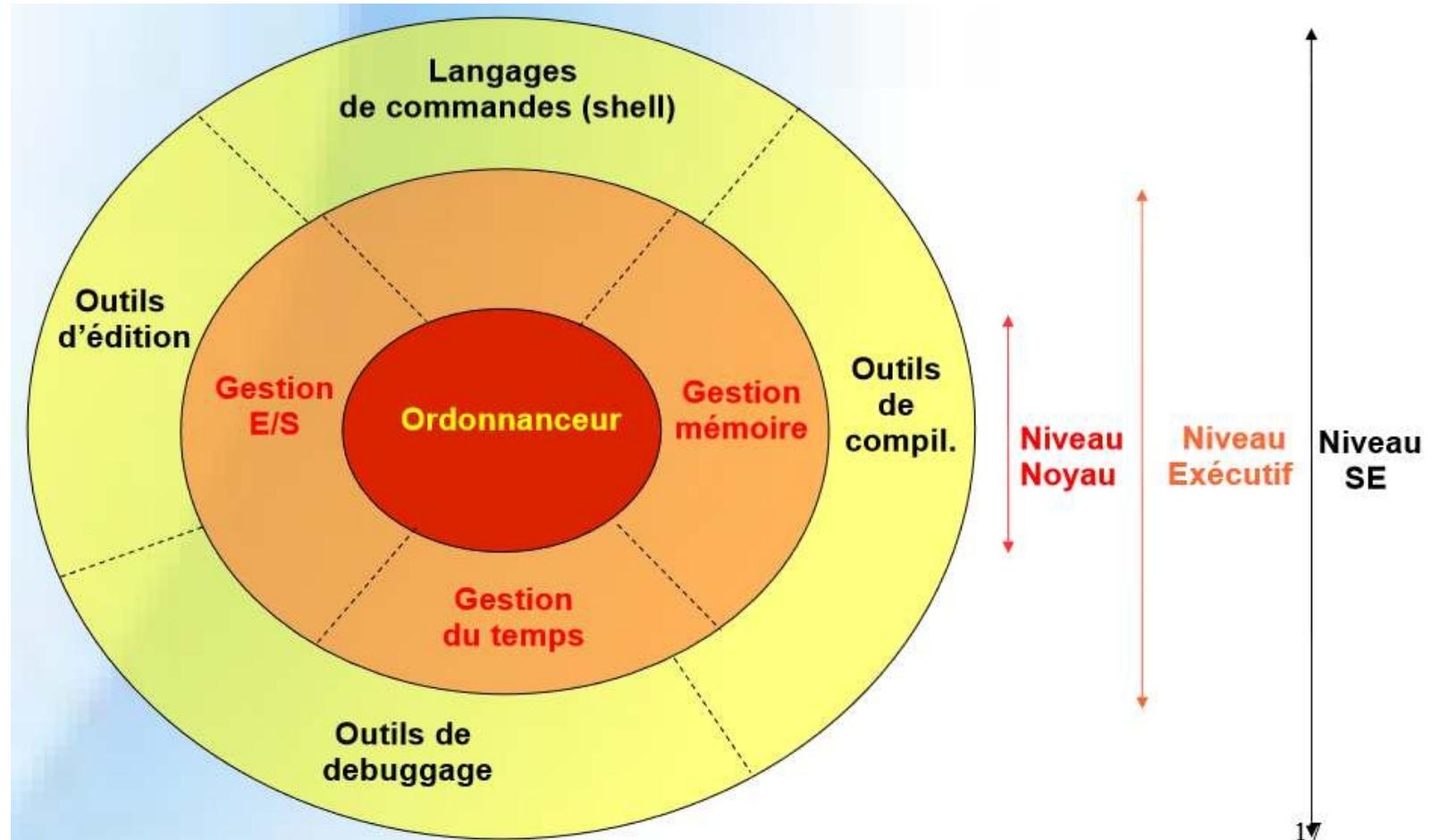
# Les systèmes d'exploitation

Filière: Génie Informatique  
S2

# **Chapitre 3**

# **La gestion des fichiers sous Linux**

## Introduction



## Introduction

Sur un système Linux, tout est fichier.

Cela peut sembler évident à première vue. Un document texte est évidemment un fichier, tout comme un document OpenOffice, une image, une vidéo ou un MP3.

Mais qu'en est-il des répertoires ? Il s'agit aussi de fichiers, d'un type particulier, qui contiennent des informations sur d'autres fichiers.

Les lecteurs de disques sont des gros fichiers.

Les connexions réseau sont des fichiers.

Même les processus en cours d'exécution sont des fichiers. Bref, sous Linux... **tout est fichier**.

## Introduction

Sous UNIX, tout élément est représenté sous forme de fichier  
4 types de fichiers :

- **Ordinaire** : données, programme
- **Répertoire** : contient d'autres données ou répertoires
- **Lien symbolique** : pointe vers un autre fichier
- **Spécial** : permet l'accès à un périphérique

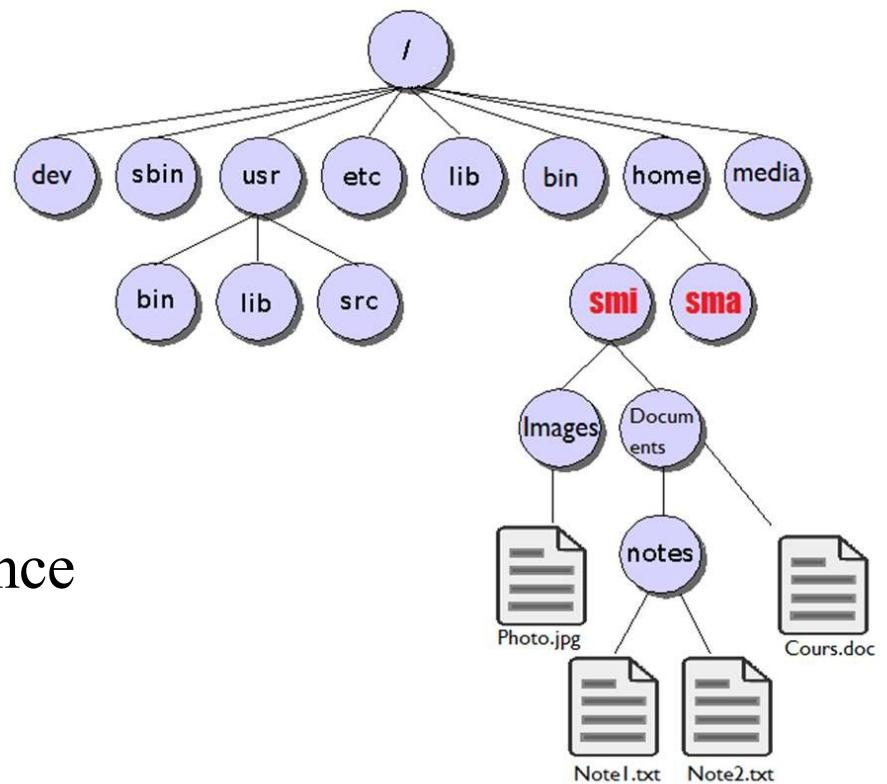
Chaque fichier est caractérisé par son nom, sa taille, ses droits d'accès, son propriétaire, ses dates de création, de modification...  
Structure arborescente de fichiers

## L'arborescence

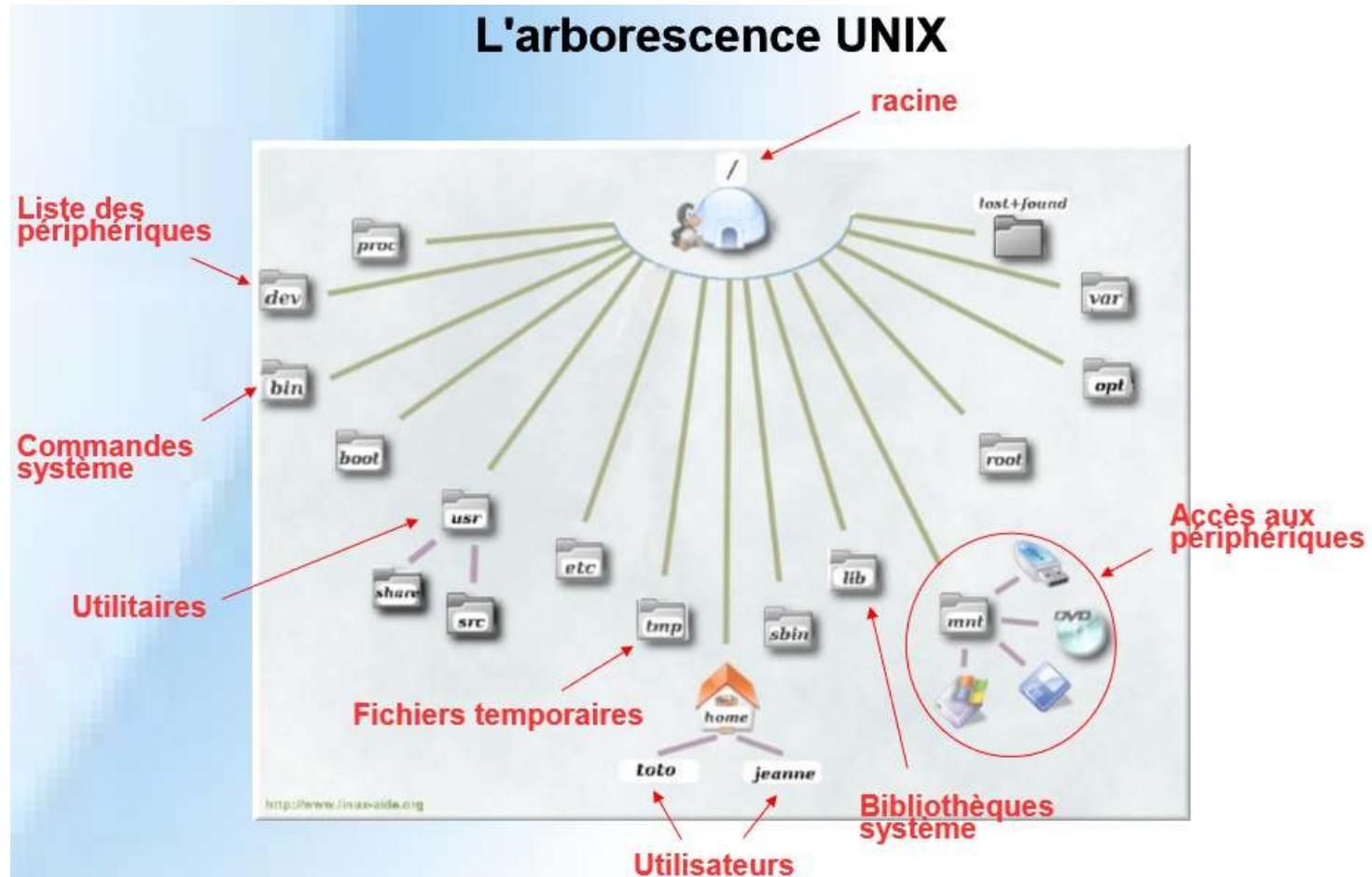
Le principe de l'arborescence sous Linux est totalement différent de celui de Windows.

Le principe est d'avoir une arborescence qui ne dépend pas des disques durs et de leurs partitions.

/ : est la racine de cette arborescence  
Les nœuds sont les répertoires  
Les fichiers sont les feuilles



## Introduction



## Chemin absolu/relatif

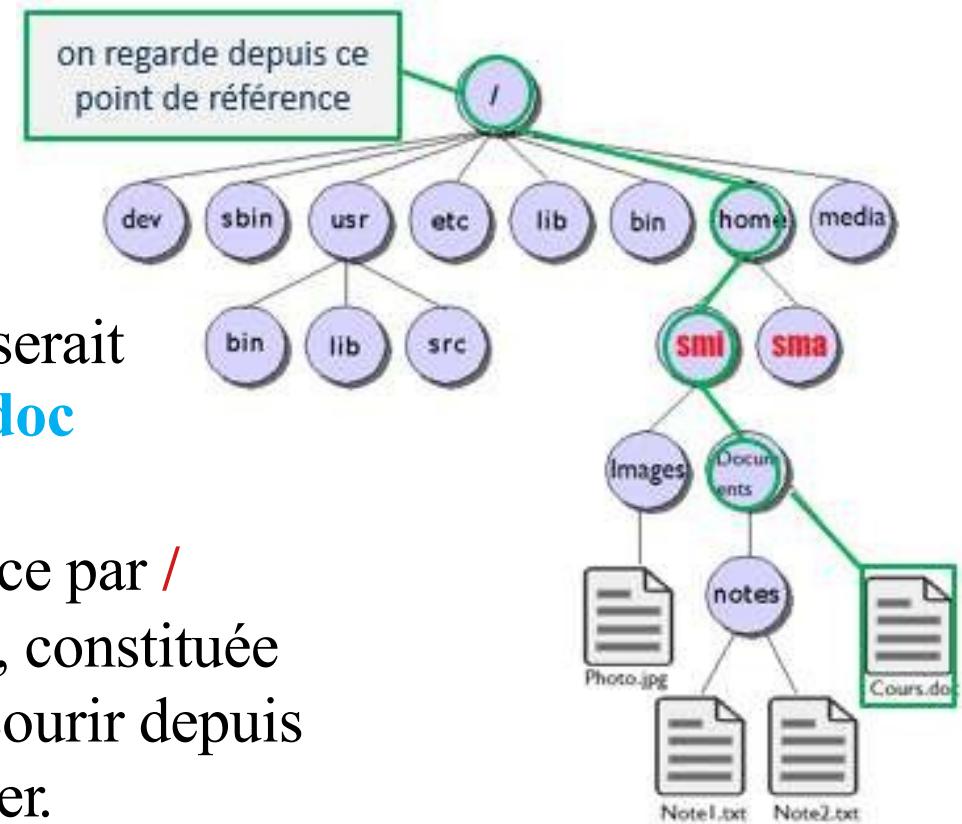
Selon ce schéma, 2 désignations possibles du chemin vers un fichier :

### Le chemin absolu

*Exemple : Cours.doc*

Partant de la racine, le chemin serait  
**/home/smi/Documents/Cours.doc**

Si le nom de répertoire commence par /  
il s'agit d'une référence absolue, constituée  
d'une liste des répertoires à parcourir depuis  
la racine / pour accéder au fichier.



## Chemin absolu/relatif

Selon ce schéma, 2 désignations possibles du chemin vers un fichier :

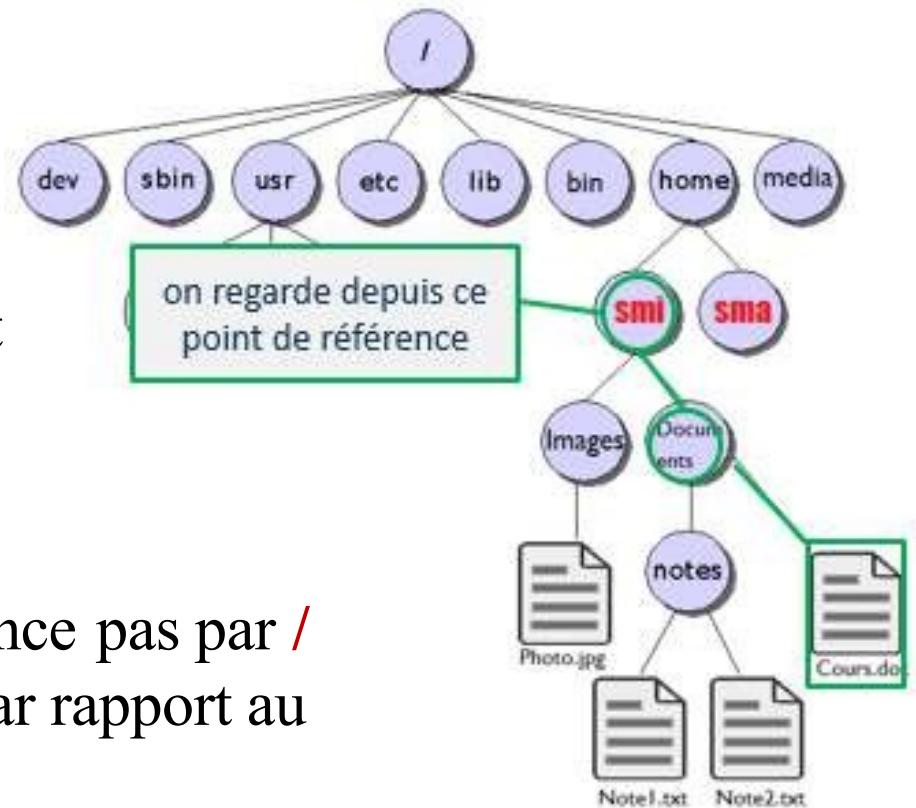
### Le chemin relatif

*Exemple : Cours.doc*

Le chemin relatif à l'emplacement où on se trouve :

**Documents/Cours.doc**

Si le nom de répertoire ne commence pas par / il s'agit d'une référence relative par rapport au répertoire courant.



## Chemin absolu/relatif

Selon ce schéma, 2 désignations possibles du chemin vers un fichier :

### Le chemin relatif

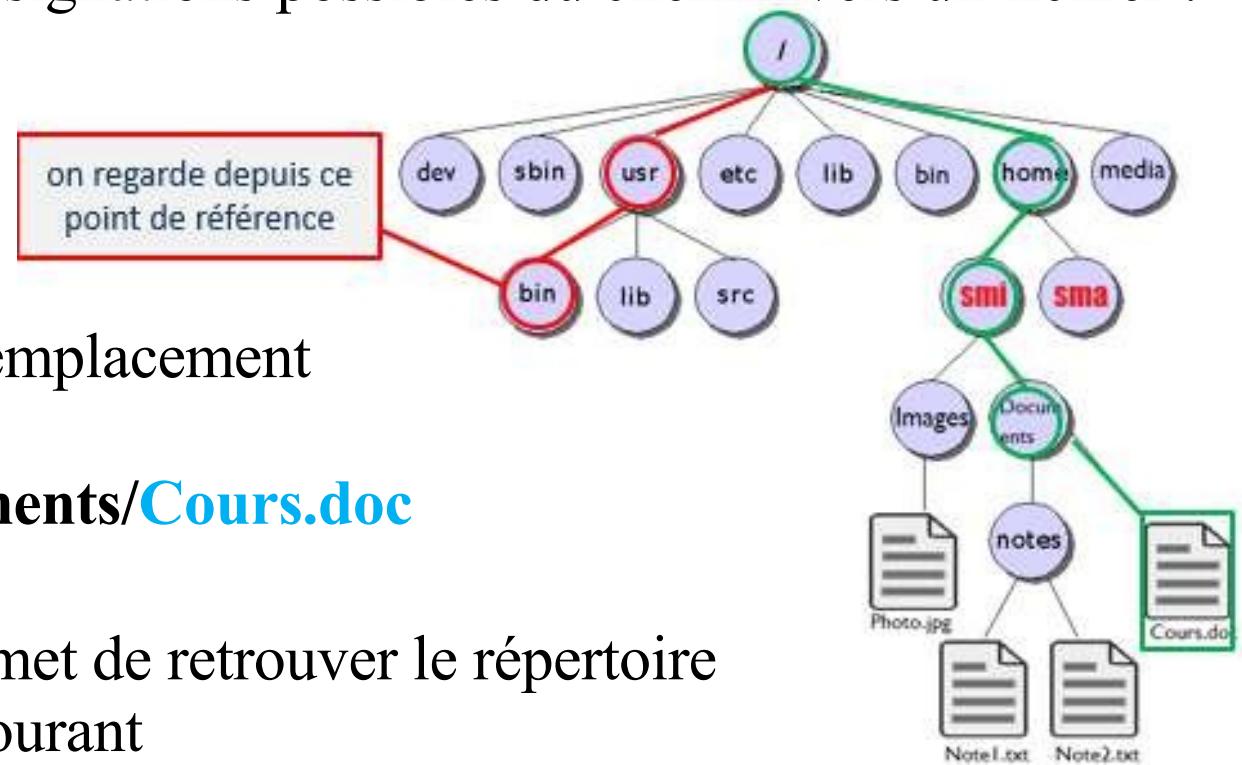
*Exemple : Cours.doc*

on regarde depuis ce point de référence

Le chemin relatif à l'emplacement où on se trouve :

**../.. /home/smi/Documents/Cours.doc**

L'utilisation de **..** Permet de retrouver le répertoire parent au répertoire courant

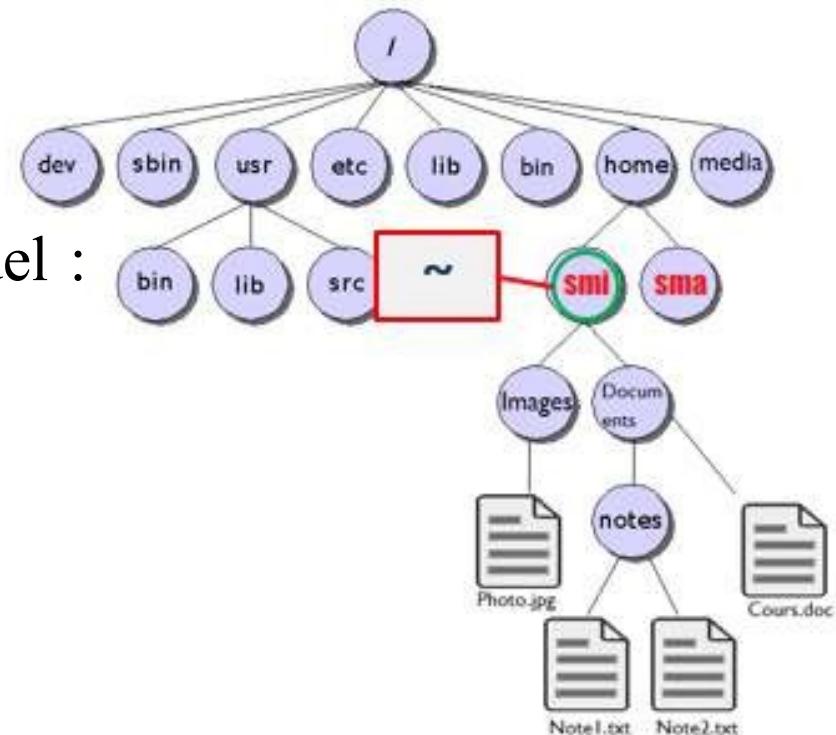


## Chemin du dossier personnel

L'opérateur `~` (tilde) : Il permet de se positionner facilement et directement dans le dossier personnel sans taper son chemin.

### Le répertoire personnel

Sur les systèmes d'exploitation de type Unix, tilde `~` indique souvent le répertoire `home` de l'utilisateur actuel : *par exemple*, si le répertoire `home` de l'utilisateur actuel est `/home/smi` on peut utiliser la commande `cd ~` à la place de `cd /home/smi` ou `cd $HOME`



## Chemin du dossier personnel

### Afficher le dossier courant

*Syntaxe : pwd [options]*

(Eng – print working directory)

La commande pwd indique votre position actuelle au sein du système de fichiers. Elle permet de savoir où vous êtes dans le système de fichiers. La commande pwd n'a pas d'arguments, il suffit de taper pwd sur le Shell :

```
smi@ubuntu-VirtualBox:~$ pwd  
/home/smi  
smi@ubuntu-VirtualBox:~$ cd Bureau  
smi@ubuntu-VirtualBox:~/Bureau$ pwd  
/home/smi/Bureau  
smi@ubuntu-VirtualBox:~/Bureau$ █
```

## Gestion des fichiers

### Lister les fichiers

*Syntaxe :* **ls** [*options*] [*paramètres*]

Permet de lister le contenu d'un dossier :

par défaut, le dossier courant si utilisée sans paramètres.  
un dossier ou plusieurs si vous spécifiez leurs chemins en paramètre

Principales options (cumulables) :

**-l** : affichage au format long des informations relatives aux objets

**-g** : affichage des groupes propriétaires des objets

**-R** : liste récursive

**-i** : affichage du **inode** des fichiers listés

**-d** : affichage des noms des objets et non de leurs contenus

**-F** : affichage des objets avec un suffixe désignant le type de l'objet

**-a** : affichage des objets dont les noms commencent par « . » (fichiers cachés)

## Gestion des fichiers

### Changer de répertoire

*Syntaxe :* **cd** [*options*] [*paramètres*]

(Eng – change directory)

Permet de se déplacer au sein du système de fichiers.

Si elle est utilisée sans argument, il vous renvoie dans votre répertoire personnel.

Le chemin de ce répertoire est nécessaire comme argument, pour déplacer vers un autre répertoire

## Gestion des fichiers

### Exemple

```
smi@ubuntu-VirtualBox:~$ ls
Bureau      examples.desktop  Modèles   Public          Vidéos
Documents    Images           Musique   Téléchargements
smi@ubuntu-VirtualBox:~$ cd Bureau
smi@ubuntu-VirtualBox:~/Bureau$ cd /
smi@ubuntu-VirtualBox:/$ ls
bin      dev  initrd.img     lib64      mnt      root    snap    sys    var
boot    etc  initrd.img.old  lost+found  opt      run    srv    tmp    vmlinuz
cdrom   home lib            media      proc    sbin    swapfile  usr
smi@ubuntu-VirtualBox:/$ cd home
smi@ubuntu-VirtualBox:/home$ ls
sma  smi  ubuntu
smi@ubuntu-VirtualBox:/home$ cd smi
smi@ubuntu-VirtualBox:~/smi$ cd ..
smi@ubuntu-VirtualBox:/home$ cd ..
smi@ubuntu-VirtualBox:/$ █
```

## Gestion des fichiers

### Création de répertoire

*Syntaxe :* **mkdir** [*options*] *paramètres*  
(Eng – make directory)

Permet de créer un répertoire (ou plusieurs) dans l'emplacement donné en *paramètres*.

Utilisée avec plusieurs arguments, elle permet de créer plusieurs répertoires dans un même niveau de l'arborescence.

**Exemples :**   **mkdir rep1 rep2 rep3**

Si on veut créer une suite de répertoires l'un inclus dans l'autre :

**mkdir -p rep1/rep2/rep3**

(similaire à : mkdir rep1 rep1/rep2 rep1/rep2/rep3)

## Gestion des fichiers

### Exemple

```
smi@ubuntu-VirtualBox:~/Bureau$ mkdir Dossier1
smi@ubuntu-VirtualBox:~/Bureau$ mkdir Dossier2,Dossier3
smi@ubuntu-VirtualBox:~/Bureau$ mkdir Dossier2;Dossier3
Dossier3 : commande introuvable
smi@ubuntu-VirtualBox:~/Bureau$ mkdir Dossier3; mkdir Dossier4
smi@ubuntu-VirtualBox:~/Bureau$ mkdir rep1 rep2 rep3
smi@ubuntu-VirtualBox:~/Bureau$ mkdir D1/D2/D3
mkdir: impossible de créer le répertoire «D1/D2/D3»: Aucun fichier ou dossier de
ce type
smi@ubuntu-VirtualBox:~/Bureau$ mkdir -p D1/D2/D3
smi@ubuntu-VirtualBox:~/Bureau$ █
```

## Gestion des fichiers

### Supprimer un répertoire

*Syntaxe :* **rmdir** [*options*] paramètres  
(Eng – remove directory)

Permet de supprimer un répertoire (ou plusieurs) dont l'emplacement est donné en paramètres.

Utilisée avec plusieurs arguments, elle permet de supprimer plusieurs répertoires dans un même niveau de l'arborescence.

**Exemples :**    **rmdir** rep1 rep2 rep3

Si on veut supprimer une suite de répertoires l'un inclus dans l'autre :

**rmdir -p** rep1/rep2/rep3

## Gestion des fichiers

### Exemple

```
smi@ubuntu-VirtualBox:~/Bureau$ rmdir Dossier1
smi@ubuntu-VirtualBox:~/Bureau$ rmdir Dossier2,Dossier3
smi@ubuntu-VirtualBox:~/Bureau$ rm Dossier2; rm Dossier3; rm Dossier4
smi@ubuntu-VirtualBox:~/Bureau$ rmdir Dossier2; rmdir Dossier3; rmdir Dossier4
smi@ubuntu-VirtualBox:~/Bureau$ rmdir rep1 rep2 rep3
smi@ubuntu-VirtualBox:~/Bureau$ rmdir D1
rmdir: impossible de supprimer 'D1': Le dossier n'est pas vide
smi@ubuntu-VirtualBox:~/Bureau$ rmdir -p D1
rmdir: impossible de supprimer 'D1': Le dossier n'est pas vide
smi@ubuntu-VirtualBox:~/Bureau$ rmdir -p D1/D2/D3
smi@ubuntu-VirtualBox:~/Bureau$ █
```

Pour forcer la suppression d'un dossier non vide

```
smi@ubuntu-VirtualBox:~/Bureau$ rm -Rf D1
```



## Gestion des fichiers

### Création et édition de fichiers

*Syntaxe : gedit [options] [paramètres]*

(Eng – text editor for the GNOME Desktop)

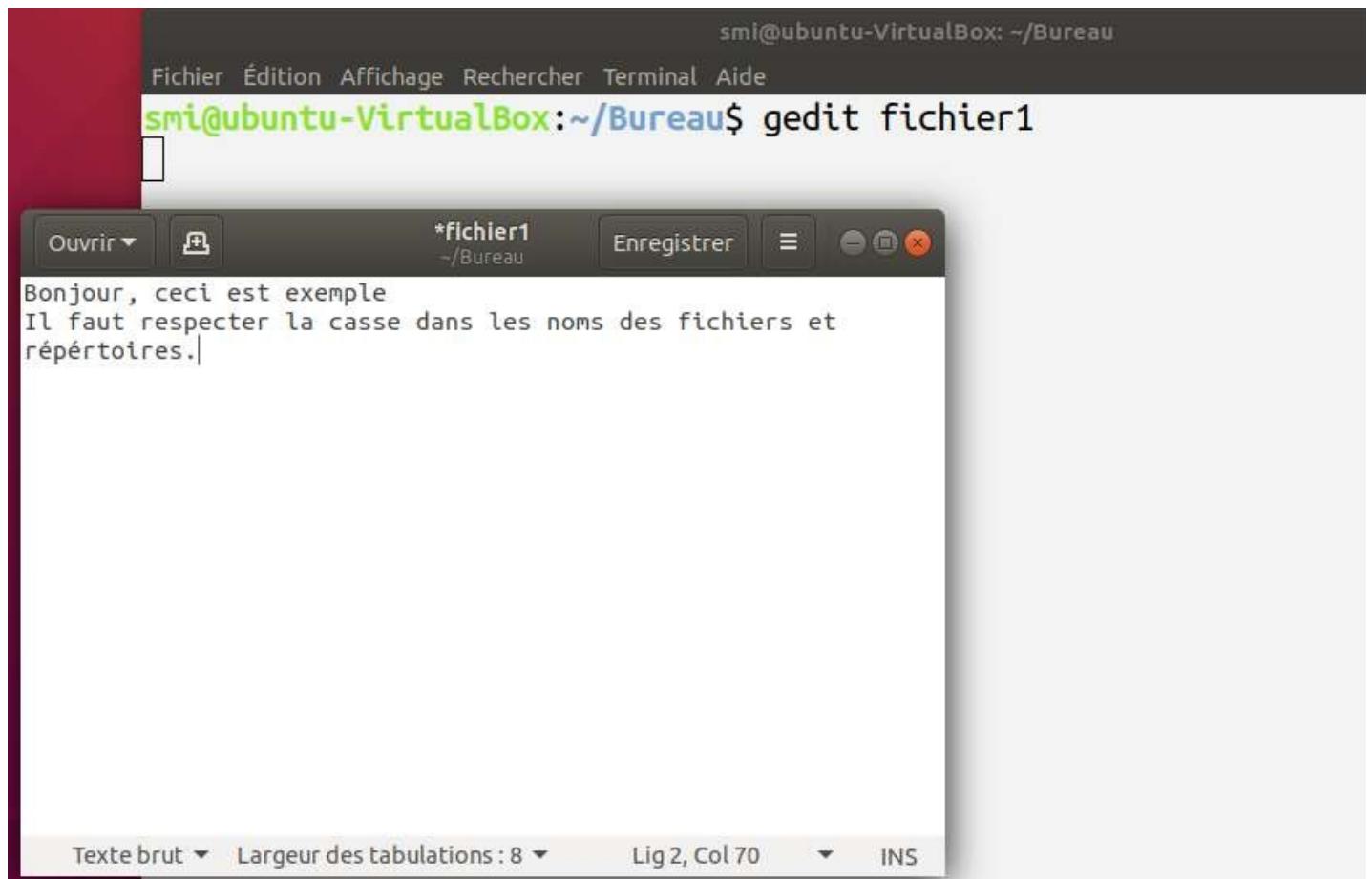
Permet de créer un fichier et d'éditer son contenu.

**gedit** est l'éditeur par défaut et officiel de l'environnement de travail GNOME. Il est facile et simple à utiliser. Son mode d'exécution est graphique. Il existe d'autre éditeur en ligne de commandes (comme Vi, Vim, nano, Emacs...) que nous rencontrerons dans la suite de ce cours.

Si utiliser sans paramètres, cela permet d'éditer un fichier sont qu'il soit créé physiquement, en attendant de l'enregistrer à partir de l'éditeur.

## Gestion des fichiers

### Exemple



## Gestion des fichiers

### Afficher le contenu d'un fichier

Syntaxe : **cat** [*options*] [*paramètres*]

(Eng – concatenate files and print on the standard output)

Permet d'afficher le contenu d'un fichier (ou plusieurs) en ligne.

Si utiliser avec l'option -n, cela permet d'énumérer les lignes du fichier.

```
smi@ubuntu-VirtualBox:~/Bureau$ cat fichier1
Bonjour, ceci est exemple
Il faut respecter la casse dans les noms des fichiers et répertoires.
smi@ubuntu-VirtualBox:~/Bureau$ cat -n fichier1
    1 Bonjour, ceci est exemple
    2 Il faut respecter la casse dans les noms des fichiers et répertoires.
smi@ubuntu-VirtualBox:~/Bureau$ █
```

## Gestion des fichiers

### Supprimer un fichier

*Syntaxe : rm [options] paramètres*

(Eng – remove files or directories)

Permet de supprimer un ou plusieurs fichiers. Par défaut, elle ne permet pas de supprimer les répertoires. Mais :

Avec l'option `-r` cela est possible.

Elle peut supprimer toute une arborescence en utilisant l'option.

**Dangereuse?!?!** Sans confirmation, suppression définitive.

L'option `-i` exécute la tâche d'une manière interactif ; demande la confirmation de chaque opération.

## Gestion des fichiers

### Exemple

```
smi@ubuntu-VirtualBox:~/Bureau$ mkdir -p D1/D2
smi@ubuntu-VirtualBox:~/Bureau$ rm D3
rm: impossible de supprimer 'D3': Aucun fichier ou dossier de ce type
smi@ubuntu-VirtualBox:~/Bureau$ rm -f D3
smi@ubuntu-VirtualBox:~/Bureau$ rm -r D3
rm: impossible de supprimer 'D3': Aucun fichier ou dossier de ce type
smi@ubuntu-VirtualBox:~/Bureau$ rm -r D1
smi@ubuntu-VirtualBox:~/Bureau$ mkdir -p D1/D2
smi@ubuntu-VirtualBox:~/Bureau$ rm -i D1
rm: impossible de supprimer 'D1': est un dossier
smi@ubuntu-VirtualBox:~/Bureau$ rm -ri D1
rm : descendre dans le répertoire 'D1' ? o
rm : supprimer 'D1/D2' du type répertoire ? o
rm : supprimer 'D1' du type répertoire ? o
smi@ubuntu-VirtualBox:~/Bureau$ █
```

## Gestion des fichiers

### Copier un fichier

*Syntaxe :* **cp** [*options*] paramètres

(Eng – copy files and directories)

Permet de copier une source vers une destination. Elle est valable pour copier des fichiers et des répertoires. Exemples d'utilisation.  
Copier un fichier vers un autre fichier:

**cp** *fich1* *fich2*

Copier des fichiers dans un répertoire :

**cp** *fich1* *fich2* *rep\_fich*

## Gestion des fichiers

### Exemple

```
smi@ubuntu-VirtualBox:~/Bureau$ gedit fichier1 fichier2
smi@ubuntu-VirtualBox:~/Bureau$ cp fichier2 fichier2 Dossier1
cp: avertissement : le fichier source 'fichier2' est mentionné plusieurs fois
smi@ubuntu-VirtualBox:~/Bureau$ cp fichier1 fichier2 Dossier1
smi@ubuntu-VirtualBox:~/Bureau$ ls Dossier1
fichier1 fichier2
smi@ubuntu-VirtualBox:~/Bureau$ █
```

## Gestion des fichiers

### Déplacer (renommer) un fichier

*Syntaxe :* **mv** [*options*] paramètres  
(Eng – move (rename) files)

Permet de déplacer une source vers un répertoire de destination. Elle est valable pour déplacer des fichiers et des répertoires.

Elle peut aussi être utilisée pour renommer un fichier

#### Exemples d'utilisation :

```
smi@ubuntu-VirtualBox:~/Bureau$ mv fichier1 Dossier1
smi@ubuntu-VirtualBox:~/Bureau$ cp fichier2 fichier1
smi@ubuntu-VirtualBox:~/Bureau$ mv fichier1 fichier2 Dossier1
smi@ubuntu-VirtualBox:~/Bureau$ █
```

## Gestion des fichiers

### Création d'un lien physique vers un fichier

Crée un fichier dans le dossier personnel :

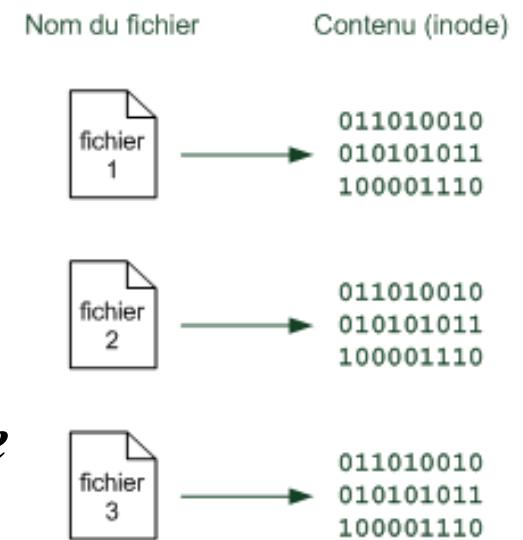
**touch fichier1**

Affichant ces propriétés étendues ( -i pour faire apparaître le inode de ce fichier) : **ls -li fichier1**

Nous constatons que ce fichier a pour **inode** 395217, et ne possède qu'un seul lien.

On peut créer un nouveau lien sur ce fichier éventuellement dans un répertoire différent si ce répertoire est sur la même partition que le fichier **fichier1**, grâce à la commande **ln** (link) dont un exemple d'utilisation :

**ln fichier1 Documents/lien\_fichier1**



## Gestion des fichiers

### Exemple

```
smi@ubuntu-VirtualBox:~/Bureau$ touch fichier1
smi@ubuntu-VirtualBox:~/Bureau$ ls -li fichier1
395217 -rw-r--r-- 1 smi smi 0 sept. 26 01:32 fichier1
smi@ubuntu-VirtualBox:~/Bureau$ ln fichier1 Lien1_fichier1
smi@ubuntu-VirtualBox:~/Bureau$ ln fichier1 ../Documents/Lien1_fichier1
smi@ubuntu-VirtualBox:~/Bureau$ ls -li fichier1
395217 -rw-r--r-- 3 smi smi 0 sept. 26 01:32 fichier1
smi@ubuntu-VirtualBox:~/Bureau$ ls -li Lien1_fichier1
395217 -rw-r--r-- 3 smi smi 0 sept. 26 01:32 Lien1_fichier1
smi@ubuntu-VirtualBox:~/Bureau$ ls -li ../Documents/Lien1_fichier1
395217 -rw-r--r-- 3 smi smi 0 sept. 26 01:32 ../Documents/Lien1_fichier1
smi@ubuntu-VirtualBox:~/Bureau$ gedit fichier1
smi@ubuntu-VirtualBox:~/Bureau$ cat ../Documents/Lien1_fichier1
Contenu de fichier1
smi@ubuntu-VirtualBox:~/Bureau$
```



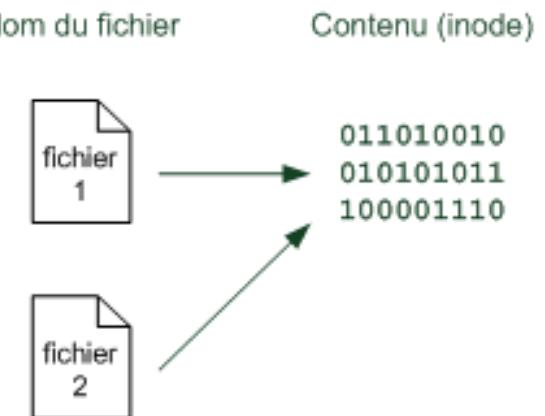
## Gestion des fichiers

### Création d'un lien symbolique vers un fichier

Lorsque l'on veut lier un fichier par l'intermédiaire d'un lien ne se trouvant pas sur la même partition que le fichier lui-même, il est nécessaire de passer par l'utilisation d'un lien dit symbolique.

**ln -s fichier1 Document/LienSymbolique\_fichier1**

```
smi@ubuntu-VirtualBox:~/Bureau$ ln -s fichier1 LienSymbolique_fichier1
smi@ubuntu-VirtualBox:~/Bureau$ ls -li fichier1
395217 -rw-r--r-- 3 smi smi 20 sept. 26 01:38 fichier1
smi@ubuntu-VirtualBox:~/Bureau$ ls -li LienSymbolique_fichier1
425715 lrwxrwxrwx 1 smi smi 8 sept. 26 01:55 LienSymbolique_fichier1 -> fichier1
```





## Gestion des fichiers

### Création d'un lien symbolique vers un fichier

La plupart des opérations sur le lien symbolique est effectuée sur le fichier sur lequel il pointe. Par contre la suppression de ce lien obéit aux règles suivantes :

- La commande rm supprime le lien symbolique lui-même (qui est un fichier à part entière) et n'a pas d'influence sur le fichier auquel il se réfère .
- Par conséquent, si on supprime le fichier, le lien symbolique existe toujours et pointe sur un fichier qui n'existe pas.
- La taille du liens ne dépend pas de la taille fichier pointé, mais plutôt de la taille qui correspond aux nombres de caractères du chemin du fichier auquel il se réfère.
- Enfin la dernière particularité du lien symbolique (qui le distingue du lien physique) provient du fait qu'il est possible de créer un tel lien sur un répertoire.

## Gestion des fichiers

### Lire les n premières lignes d'un fichier

**Syntaxe :** `head -n fichier`

Permet d'afficher les n premières lignes d'un fichier

**Exemples :** `head -5 /home/smi/Bureau/fichier1`

Affiche les cinq premières lignes du fichier « `fichier1` »

### Lire les n dernières lignes d'un fichier

**Syntaxe :** `tail -n fichier`

Permet d'afficher les n dernières lignes d'un fichier

**Exemples :** `tail -4 /home/smi/Bureau/fichier1`

Affiche les quatre dernières lignes du fichier « `fichier1` »

## Gestion des fichiers

### Chercher une chaîne de caractères

*Syntaxe : grep [options] chaîne\_à\_chercher fichiers*

Permet de chercher une chaîne de caractères dans un fichier

**Pour afficher le numéro de ligne où se trouve la chaîne(option -n)**

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ grep -n "Système" Fichier3
1:Système d'exploitation
2:Système d'information
4:Restauration du Système
```

**Pour ne pas respecter la casse (option -i)**

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ grep -i "Système" Fichier3
Système d'exploitation
Système d'information
Fichier système
Restauration du Système
```

## Gestion des fichiers

### Chercher une chaîne de caractères

Pour afficher le fichier où se trouve la chaîne (option -H)

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ grep -H "Système" Fichier3
Fichier3:Système d'exploitation
Fichier3:Système d'information
Fichier3:Restauration du Système
```

Pour afficher seulement le fichier de la chaîne (option -l)

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ grep -l "Système" Fichier3
Fichier3
```

Pour afficher le nombre de fois où chaîne se présente (option -c)

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ grep -c "Système" Fichier3
3
```

## Gestion des fichiers

### Chercher une chaîne de caractères

Pour chercher récursivement dans tous les fichiers d'un dossier

```
ubuntu@ubuntu-VirtualBox:~$ grep -r "Système" Bureau
Bureau/SousDossier/Fichier4:Données Système
Bureau/SousDossier/Fichier4:Système de refroidissement
Bureau/Fichier3:Système d'exploitation
Bureau/Fichier3:Système d'information
Bureau/Fichier3:Restauration du Système
```

Pour afficher les lignes qui ne contiennent pas la chaîne(option –v)

```
ubuntu@ubuntu-VirtualBox:~$ grep -nv "Système" Bureau/Fichier3
3:Fichier système
```

## Gestion des fichiers

### Trier un fichier de données

*Syntaxe : sort [options] fichier*

Soit le fichier « *noms.txt* » à droite:

Pour trier les noms

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ sort noms.txt
```

Brahim

Ikram

Karim

Laila

Mohamed

Saad

Salma

GNU nano 2.9.3

Mohamed

Karim

Salma

Ikram

Saad

Laila

Brahim █

Pour trier les noms et mettre le résultat dans un autre fichier

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ sort -o nomsTriés.txt noms.txt
```

## Gestion des fichiers

**Trier un fichier de données**

Soit le fichier « noms.txt » à droite:

**Pour trier les noms en ordre inverse**

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ sort -r noms.txt
```

Salma

Saad

Mohamed

Laila

Karim

Ikram

Brahim

GNU nano 2.9.3

Mohamed

Karim

Salma

Ikram

Saad

Laila

Brahim █

**Pour trier les noms aléatoirement**

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ sort -R noms.txt
```

## Gestion des fichiers

**Trier un fichier de données**

**Soit le fichier « noms.txt » à droite:**

**Pour trier les nombres**

~~ubuntu@ubuntu-VirtualBox:~/Bureau\$ sort nombres.txt~~

~~12  
24  
34  
5  
57  
76  
90~~

GNU nano 2.9.3

24  
12  
5  
34  
57  
90  
76  
█

**C'est faux**

## Gestion des fichiers

### Trier un fichier de données

Soit le fichier « noms.txt » à droite:

Pour trier les nombres avec l'option **-n**

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ sort -n nombres.txt
```

```
5  
12  
24  
34  
57  
76  
90
```

GNU nano 2.9.3

24  
12  
5  
34  
57  
90  
76  
█

Maintenant oui

## Gestion des fichiers

### Compter ( La commande WC)

*Syntaxe : wc [options] fichier*

Soit le fichier « noms.txt » à droite:

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ wc noms.txt
 7  7 44 noms.txt
```

Ces trois nombres signifient, dans l'ordre :

- 1.le nombre de lignes (ici 7).
- 2.le nombre de mots (ici 7).
- 3.le nombre d'octets (ici 44)

NB: pour le nombre d'octets, n'oubliez pas de compter les octets du caractère « retour à la ligne »

GNU nano 2.9.3

Mohamed  
Karim  
Salma  
Ikram  
Saad  
Laila  
Brahim



## Gestion des fichiers

GNU nano 2.9.3

### Compter ( La commande WC)

Soit le fichier « noms.txt » à droite:

Pour afficher uniquement le nombre de lignes

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ wc -l noms.txt  
7 noms.txt
```

Mohamed  
Karim  
Salma  
Ikram  
Saad  
Laila  
Brahim

Pour afficher uniquement le nombre de mots

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ wc -w noms.txt  
7 noms.txt
```

Pour afficher uniquement le nombre d'octets

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ wc -c noms.txt  
44 noms.txt
```

## Gestion des fichiers

### Supprimer les doublons d'un fichier

Soit le fichier « *noms\_doublons.txt* » à droite:

Pour supprimer les doublons on applique la commande **Uniq** sur un fichier **Trié**

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ uniq noms_doublons_trie.txt
```

```
Brahim  
Ikram  
Karim  
Laila  
Mohamed  
Saad  
Salma
```

GNU nano 2.9.3

```
Brahim  
Brahim  
Ikram  
Karim  
Laila  
Mohamed  
Mohamed  
Saad  
Saad  
Salma
```

## Gestion des fichiers

### Supprimer les doublons d'un fichier

Soit le fichier « *noms\_doublons.txt* » à droite:

Pour supprimer les doublons et mettre le résultat dans un autre fichier

GNU nano 2.9.3

Brahim  
Brahim  
Ikram  
Karim  
Laila

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ uniq noms_doublons_trie.txt noms_sans_doublons.txt
ubuntu@ubuntu-VirtualBox:~/Bureau$ cat noms_sans_doublons.txt
Brahim
Ikram
Karim
Laila
Mohamed
Saad
Salma
```

## Gestion des fichiers

### Supprimer les doublons d'un fichier

Pour compter le nombre d'occurrences de chaque nom, on utilise l'option **-c**

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ uniq -c noms_doublons_trié.txt
      2 Brahim
      1 Ikram
      1 Karim
      1 Laila
      2 Mohamed
      2 Saad
      1 Salma
```

## Gestion des fichiers

### Supprimer les doublons d'un fichier

Pour afficher seulement les noms en double, on utilise l'option **-d**

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ uniq -d noms_doublons_trié.txt
Brahim
Mohamed
Saad
```



## Gestion des fichiers

### Couper une partie d'un fichier

*Syntaxe :* **cut** [*options*] *fichier*

#### Couper selon le nombre de caractères

Si vous souhaitez conserver uniquement les caractères 2 à 5 de chaque ligne du fichier, vous taperez :

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ cut -c 2-5 noms.txt
oham
arim
alma
kram
aad
aila
rahi
oham
aad
rahi
```

## Gestion des fichiers

### Couper une partie d'un fichier

*Syntaxe :* **cut** [*options*] *fichier*

#### Couper selon le nombre de caractères

Si vous souhaitez conserver le premier jusqu'au 5 ème de chaque ligne du fichier, vous taperez :

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ cut -c -5 noms.txt
```

Si vous souhaitez conserver le 5 ème jusqu'au dernier de chaque ligne du fichier, vous taperez :

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ cut -c 5- noms.txt
```

## Gestion des fichiers

### Couper une partie d'un fichier

*Syntaxe : cut [options] fichier*

#### Couper selon un délimiteur

Soit le fichier « Etudiants.txt » à droite:

Pour avoir seulement les prénoms

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ cut -d ";" -f 1 Etudiants.txt
```

Karim

Hasnaa

Laila

Mohamed

Rachid

Karim;SELLAMI;23;15/20

Hasnaa;BAKKALI;22;16/20

Laila;MOUSSAOUI;21;14/20

Mohamed;SABIRI;24;12/20

Rachid;BOUKHARI;21;17/20

-d : indique quel est le délimiteur dans le fichier ;  
-f : indique le numéro du ou des champs à couper.

## Gestion des fichiers

### Couper une partie d'un fichier

*Syntaxe : cut [options] fichier*

#### Couper selon un délimiteur

Soit le fichier « Etudiants.txt » à droite:

Pour avoir les noms et les prénoms

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ cut -d ";" -f 2,1 Etudiants.txt
```

Pour avoir les données de la deuxième colonne jusqu'à la quatrième colonne

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ cut -d ";" -f 2-4 Etudiants.txt
```

Karim;SELLAMI;23;15/20  
Hasnaa;BAKKALI;22;16/20  
Laila;MOUSSAOUI;21;14/20  
Mohamed;SABIRI;24;12/20  
Rachid;BOUKHARI;21;17/20

## Gestion des fichiers

### Chercher un fichier (**locate**)

*Syntaxe : locate [options] chaine*

**Locate** vous donne tous les fichiers qui contiennent le mot « **chaine** » dans leur nom. Que ce soient des fichiers ou des dossiers, elle ne fait pas la différence. Elle vous donne la liste complète des fichiers qu'elle a trouvés.

```
ubuntu@ubuntu-VirtualBox: ~/Bureau$ locate noms  
/home/ubuntu/noms.txt  
/home/ubuntu/Bureau/noms.txt  
/home/ubuntu/Bureau/nomsTriés.txt
```

## Gestion des fichiers

### Chercher un fichier (**locate**)

*Exemple pratique :*

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ touch MonFichier  
ubuntu@ubuntu-VirtualBox:~/Bureau$ locate MonFichier  
ubuntu@ubuntu-VirtualBox:~/Bureau$ █
```

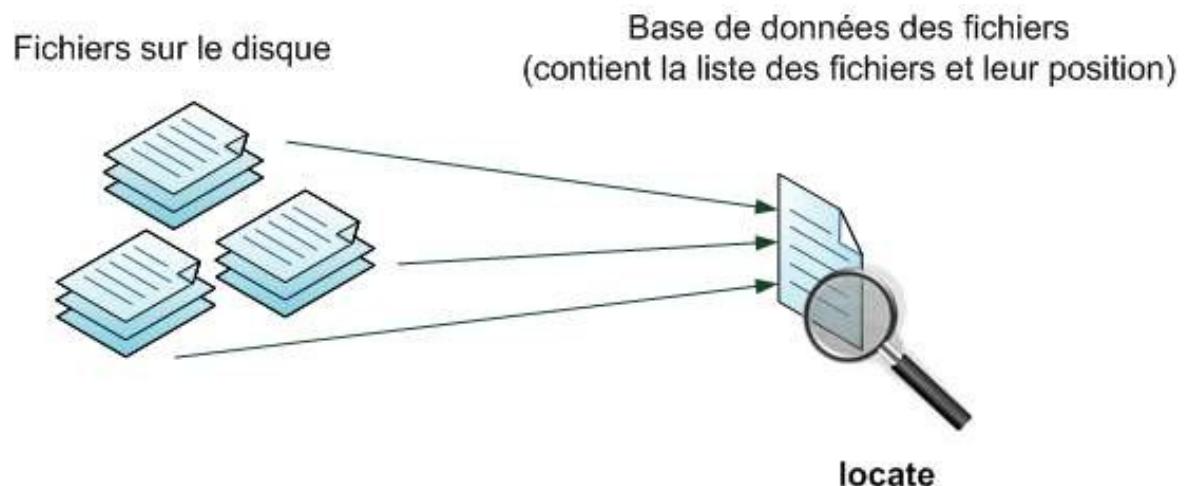
Je viens de créer des fichiers (avec la commande **touch** par exemple), et **locate** ne me renvoie aucun résultat. Pourquoi ?

la commande **locate** ne fait pas la recherche sur votre disque dur entier, mais seulement sur une base de données de vos fichiers.

## Gestion des fichiers

### Chercher un fichier (**locate**)

*Exemple pratique :*



les fichiers viennent tout juste d'être créés et n'ont pas encore été répertoriés dans la base de données. Ils ne seront donc pas découverts par **locate**.

Une fois par jour, votre système mettra à jour la base de données. Donc, si vous réessayez demain, il est probable que **locate** trouve enfin votre fichier.



## Gestion des fichiers

### Chercher un fichier (**locate**)

*Exemple pratique :*

Je dois attendre 24 heures pour mettre à jour la base de données des fichiers?

Vous pouvez forcer la commande **locate** à reconstruire la base de données des fichiers du disque dur. Cela se fait avec la commande **updatedb**, à exécuter en **root** (avec **sudo**) :

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ sudo updatedb
[sudo] Mot de passe de ubuntu :
ubuntu@ubuntu-VirtualBox:~/Bureau$ locate MonFichier
/home/ubuntu/Bureau/MonFichier
```



## Gestion des fichiers

### Chercher un fichier (**find**)

**find** est la commande de recherche par excellence pour retrouver des fichiers, mais aussi pour effectuer des opérations sur chacun des fichiers trouvés. Elle est très puissante, permet donc de faire beaucoup de choses, et par conséquent... elle est un peu complexe.

Contrairement à **locate**, **find** ne va pas lire dans une base de données mais au contraire parcourir tout votre disque dur. Cela peut être très long si vous avez plusieurs giga-octets de données !



## Gestion des fichiers

### Chercher un fichier (**find**)

La commande **find** s'utilise de la façon suivante :

**find** « où » « quoi » « que faire avec »

Seul le paramètre « quoi » est obligatoire.

**Où** : Nom du dossier dans lequel la commande va faire la recherche.

**Quoi** : C'est le fichier à rechercher. On peut rechercher un fichier par son nom, mais aussi en fonction de la date de sa création, de sa taille.

**Que faire avec** : il est possible d'effectuer des actions automatiquement sur chacun des fichiers trouvés (on parle de « post-traitement »).

## Gestion des fichiers

### Chercher un fichier (find)

#### Utilisation basique de la commande find

##### Recherche à partir du nom

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ find -name noms.txt  
./noms.txt
```

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ find /home/ubuntu/ -name noms.txt  
/home/ubuntu/Bureau/noms.txt  
/home/ubuntu/noms.txt
```

Chemin dans lequel on cherche

Par défaut, find est sensible à la casse, c'est à dire qu'il différencie les majuscules et les minuscules. Une recherche de "photo.jpg" ne permettra pas de trouver le fichier "Photo.jpg". Pour rendre find insensible à la casse, il faut utiliser le paramètre **-iname** à la place de **-name**.



## Gestion des fichiers

### Chercher un fichier (find)

#### Utilisation basique de la commande find

##### Recherche à partir de la taille

Chercher les fichier dont la taille est supérieure à 4200Ko

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ find /home/ubuntu/ -size +4200k
```

Chercher les fichier dont la taille est inférieure à 5Mo

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ find /home/ubuntu/ -size -5M
```

Chercher les fichier dont la taille est comprise entre 4Mo et 5Mo

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ find /home/ubuntu/ -size +4M -size -5M
```

Si la taille est suivie de c, la valeur est exprimée en octets.



## Gestion des fichiers

### Chercher un fichier (find)

#### Utilisation basique de la commande find

#### Recherche à partir de la date de dernier accès

Avec **-atime**, vous pouvez indiquer le nombre de jours qui vous séparent du dernier accès à un fichier.

Inférieur strictement à un jour

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ find /home/ubuntu/ -name "noms*" -atime -1
/home/ubuntu/Bureau/noms_doublons.txt
/home/ubuntu/Bureau/noms_sans_doublons.txt
/home/ubuntu/Bureau/noms_doublons_trié.txt
/home/ubuntu/Bureau/noms.txt
ubuntu@ubuntu-VirtualBox:~/Bureau$ find /home/ubuntu/ -name "noms*" -atime +0
/home/ubuntu/Bureau/nomsTriés.txt
/home/ubuntu/noms.txt
```

Fichiers commençant par ‘noms’

Supérieur strictement à un jour

## Gestion des fichiers

### **Chercher un fichier (find)**

#### **Utilisation basique de la commande find**

#### **Rechercher uniquement des répertoires ou des fichiers**

On peut aussi rechercher uniquement des répertoires ou des fichiers.

On utilise:

-type **d** : pour rechercher uniquement des répertoires (directories) ;

-type **f** : pour rechercher uniquement des fichiers (files).

## Gestion des fichiers

### Chercher un fichier (find)

#### Utilisation basique de la commande find

#### Afficher les fichiers de façon formatée

Par défaut, on liste juste les noms des fichiers trouvés. On peut cependant avec l'option -printf, qui rappellera à certains le langage C, manipuler un peu ce qui est affiché.

```
ubuntu@ubuntu-VirtualBox:~$ find -name "noms*" -printf "%p: %s ko - user: %u \n"
./Bureau/noms_doublons.txt: 64 ko - user: ubuntu
./Bureau/nomsTriés.txt: 44 ko - user: ubuntu
./Bureau/noms_sans_doublons.txt: 44 ko - user: ubuntu
./Bureau/noms_doublons_trié.txt: 64 ko - user: ubuntu
./Bureau/noms.txt: 64 ko - user: ubuntu
./noms.txt: 44 ko - user: ubuntu
```

Avec **%p**: chemin du fichier; **%s**: taille du fichier; **%u**: nom d'utilisateur



## Gestion des fichiers

### Chercher un fichier (find)

Utilisation basique de la commande find

### Supprimer les fichiers trouvés

Pour supprimer tous les fichiers trouvés

```
ubuntu@ubuntu-VirtualBox:~$ find -name "noms*" -delete
```

### Appeler une commande

Avec **-exec**, vous pouvez appeler une commande qui effectuera une action sur chacun des fichiers trouvés.

```
ubuntu@ubuntu-VirtualBox:~$ find -name "noms*" -exec cat {} \;
```

**Exercice :** essayez de regrouper tous les fichiers.jpg éparpillés dans votre répertoire home dans un dossier images.