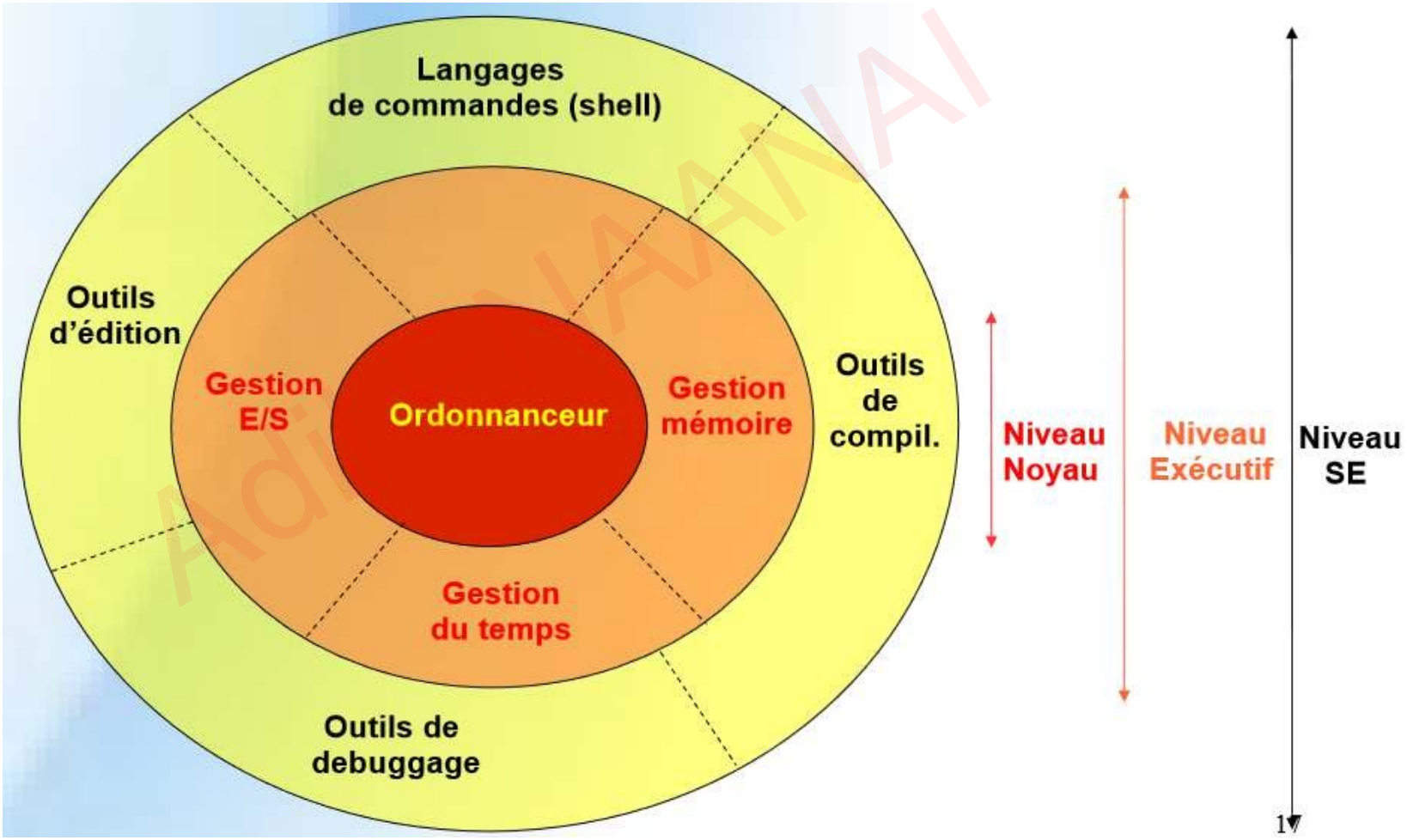
Operating systems

Adil ENAANAI
adil.enaanai@gmail.com
Computer Science Department
Faculty of Science -Tetouan-

Chapter 3

File management under Linux

Introduction



Introduction

On a Linux system, everything is a file.
This may seem obvious at first glance. A text document is obviously a file, just like an OpenOffice document, an image, a video or an MP3.
But what about directories? They're also files, a
which contain information about other files.
Disk drives are big files. Network connections
are files.
Even running processes are files. In short, **everything** in Linux **is a file**.

Introduction

Under UNIX, every element is represented as a file 4 file types :

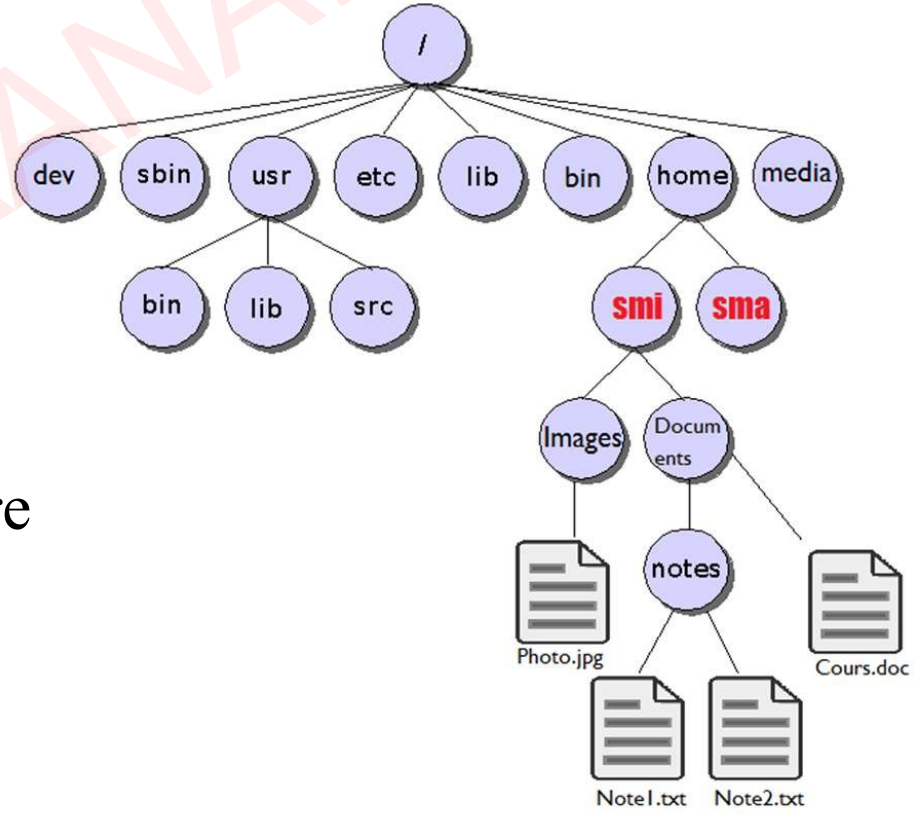
- **Ordinary**: data, program
- **Directory**: contains other data or directories
- **Symbolic link**: points to another file
- **Special**: allows access to a device

Each file is characterized by its name, size, access rights, owner, creation and modification dates, etc. Tree structure of files

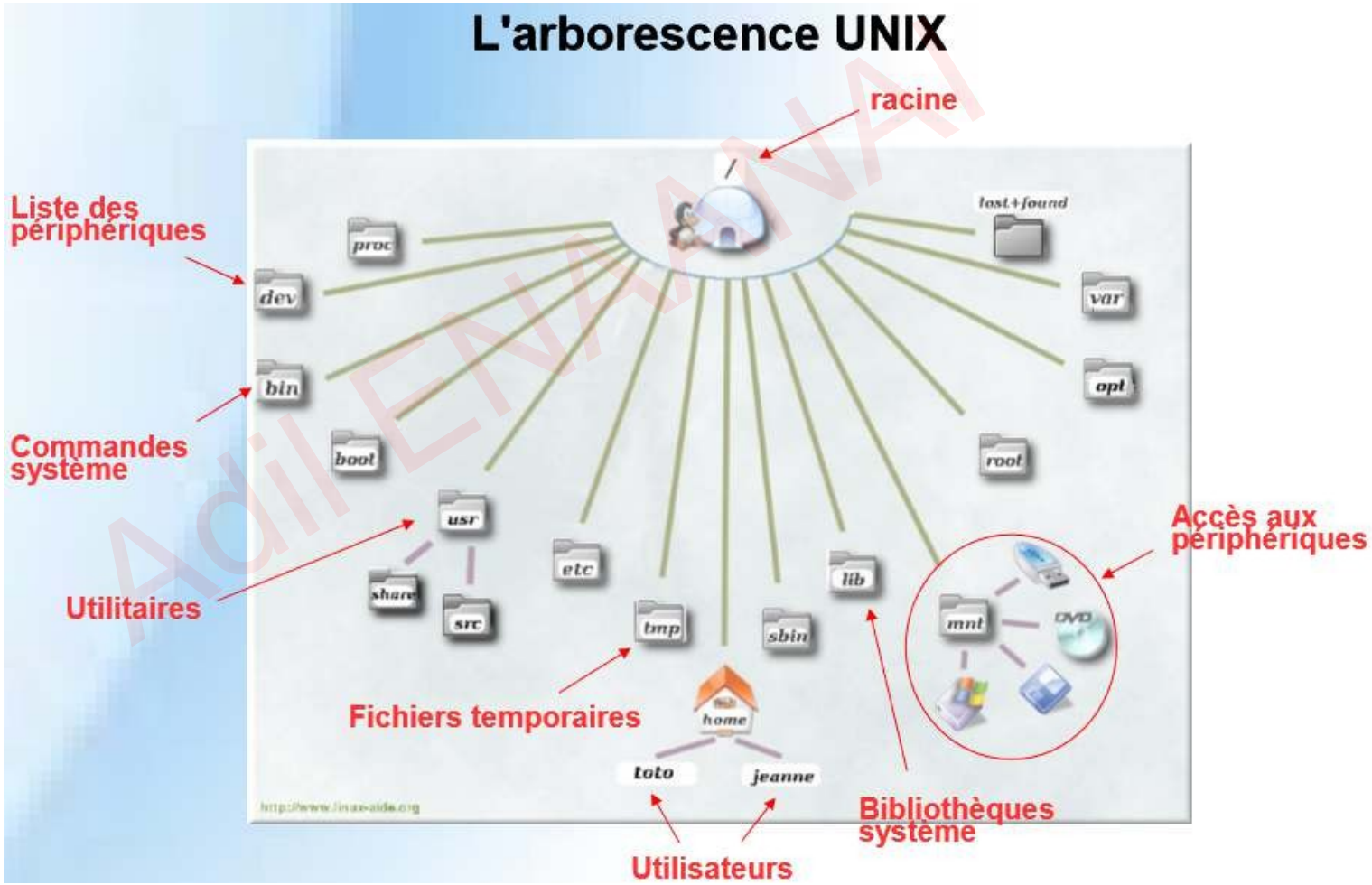
The tree structure

The tree structure principle under Linux is totally different from that of Windows.
The principle is to have a tree structure that doesn't depend on hard disks
and their scores.

/ : is the root of this tree Nodes are directories
Files are sheets

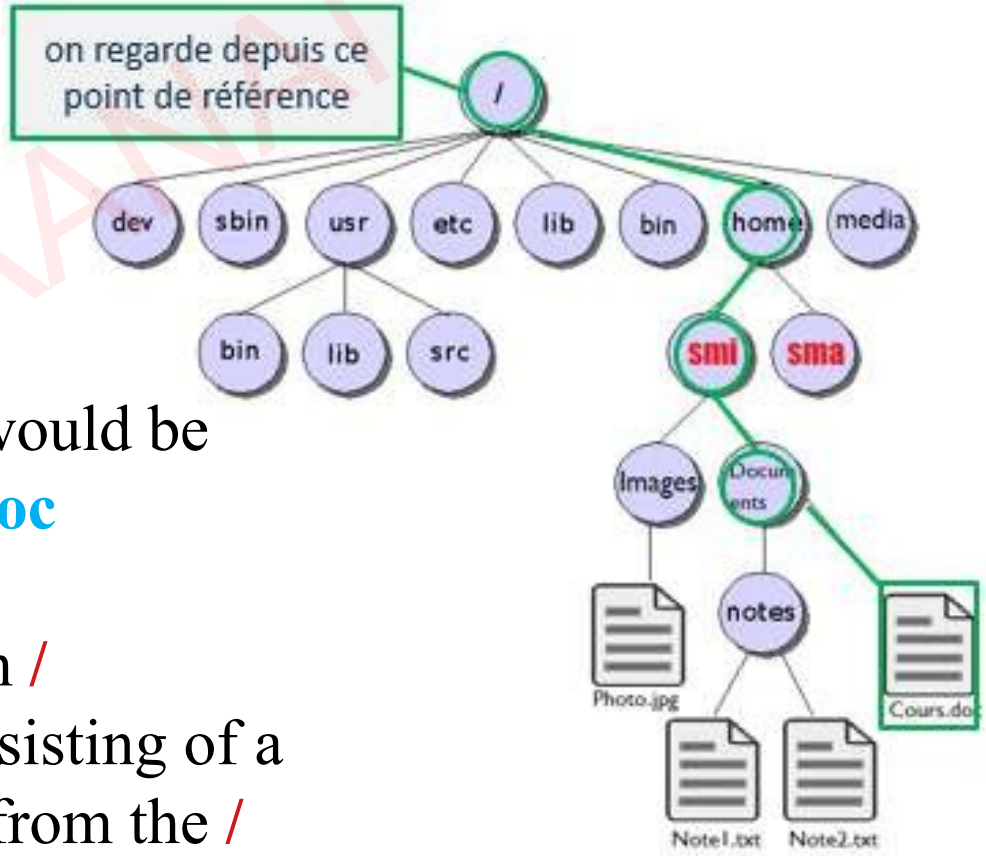


Introduction



Absolute/relative path

According to this diagram, there are 2 possible designations for the path to a file:



The absolute way

Example: Course.doc

Starting from the root, the path would be **/home/smi/Documents/Cours.doc**

If the directory name begins with /
this is an absolute reference, consisting of a list of directories to be browsed from the / root to access the file.

Absolute/relative path

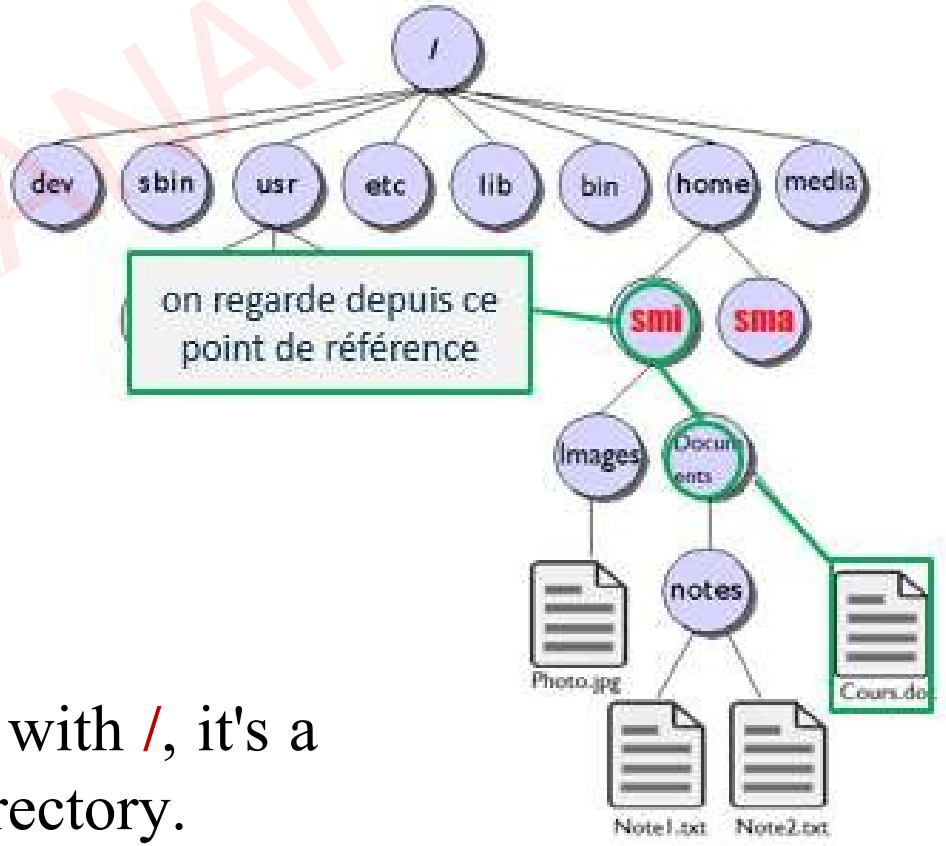
According to this diagram, there are 2 possible designations for the path to a file:

The relative path

Example: Course.doc

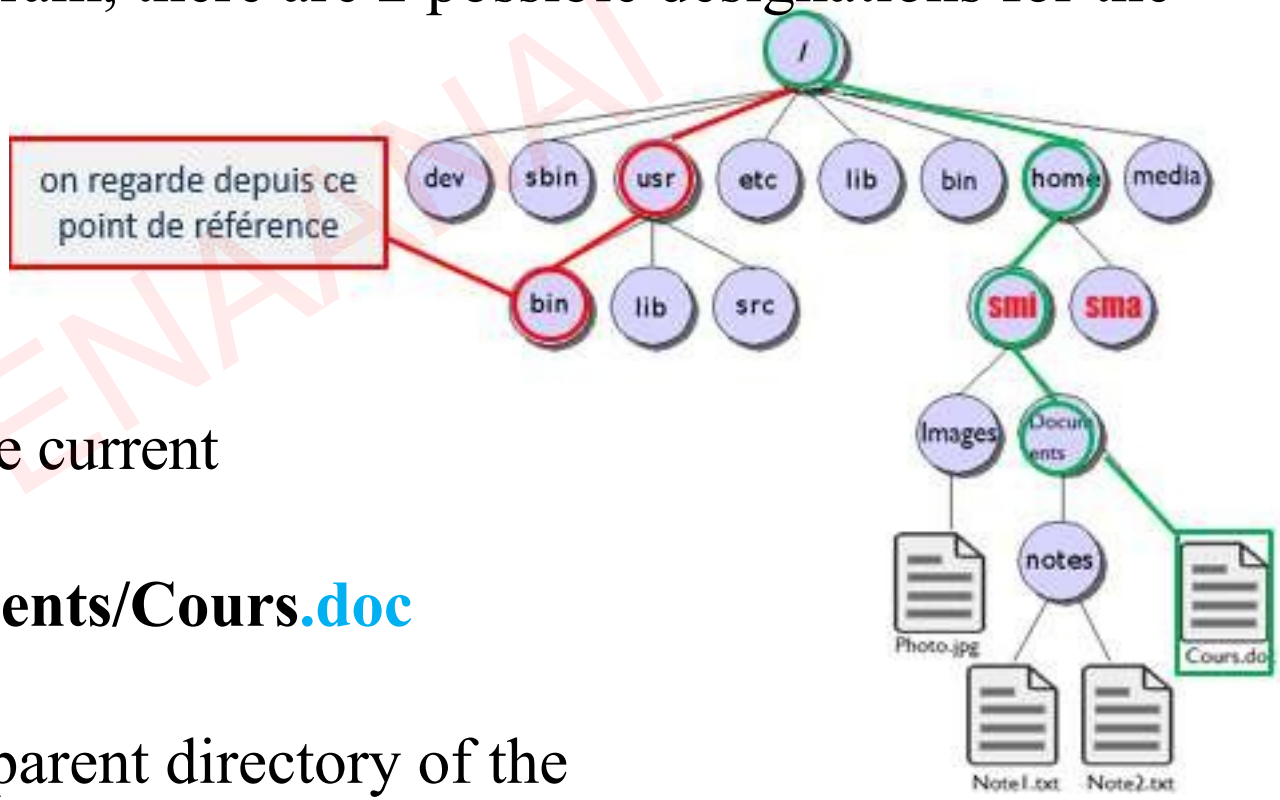
The path relative to the location where you are:
Documents/Cours.doc

If the directory name doesn't start with /, it's a relative reference to the current directory.



Absolute/relative path

According to this diagram, there are 2 possible designations for the path to a file:



The relative path

Example: Course.doc

The path relative to the current location:

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

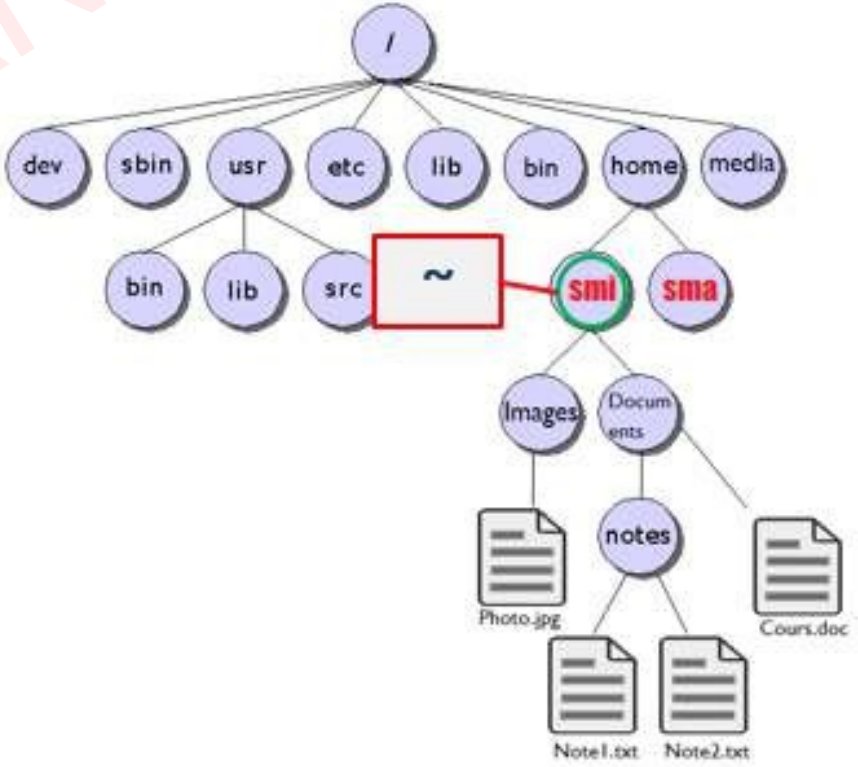
The use of . finds the parent directory of the current directory

Personal folder path

Operator ~ (tilde): Allows you to position yourself easily and directly in your personal folder without having to type your path.

The personal directory

On operating systems
type, tilde ~ often indicates
the current user's home directory: ***for example,*** if the current user's home directory is **/home/smi**
you can use the cd ~ command
instead of
cd /home/smi or cd **$HOME**



Home folder path View

current folder

Syntax: **pwd** [*options*]

(Eng - *print working directory*)

The pwd command indicates your current position in the file system. It lets you know where you are in the file system. The pwd command has no arguments, just type pwd on the Shell :

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

File management List

files

Syntax: **ls** [*options*] [parameters]

Lists the contents of a folder:

by default, the current folder if used without parameters.

one or more folders if you specify their paths in the parameter Main options (cumulative) :

-l: display object information in long format

-g: display object owner groups

-R: recursive list

-i: display the **inode** of listed files

-d: display object names rather than their contents

-F: display objects with a suffix designating the object type

-a: display objects whose names begin with ".

File management

Change directory

Syntax: **cd** [*options*] [parameters]

(Eng - change directory)

Allows you to move around the file system.

If used without argument, it returns you to your home directory.

The path of this directory is needed as an argument to move to another directory.

File management

Example

```
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:~$ cd ..
smi@ubuntu-VirtualBox:/home$ cd ..
smi@ubuntu-VirtualBox:/$
```

File management

Directory creation

Syntax: **mkdir** [*options*] parameters

(Eng - make directory)

Creates a directory (or several) in the location given as parameters.

When used with multiple arguments, it can be used to create multiple directories within a single tree level.

Examples: **mkdir** rep1 rep2 rep3

If you want to create a series of directories, one included in the other :

mkdir *-p* rep1/rep2/rep3

(similar to: mkdir rep1 rep1/rep2 rep1/rep2/rep3)

File management

Example

```
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$
```

File management Delete a

directory

Syntax: **rmdir** [*options*] parameters

(Eng - remove directory)

Allows you to delete a directory (or several) whose location is given in parameters.

When used with multiple arguments, it can be used to delete multiple directories within a single tree level.

Examples: **rmdir** rep1 rep2 rep3

If you want to delete a series of directories one included in the other :

rmdir *-p* rep1/rep2/rep3

File management

Example

```
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$
```

To force deletion of a non-empty folder

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

File management File

creation and editing

Syntax: **gedit** [*options*] [parameters]

(Eng - text editor for the GNOME Desktop)

Enables you to create a file and edit its contents.

gedit is the official default editor for the GNOME work environment.

It's easy and straightforward to use. Its execution mode is graphical.

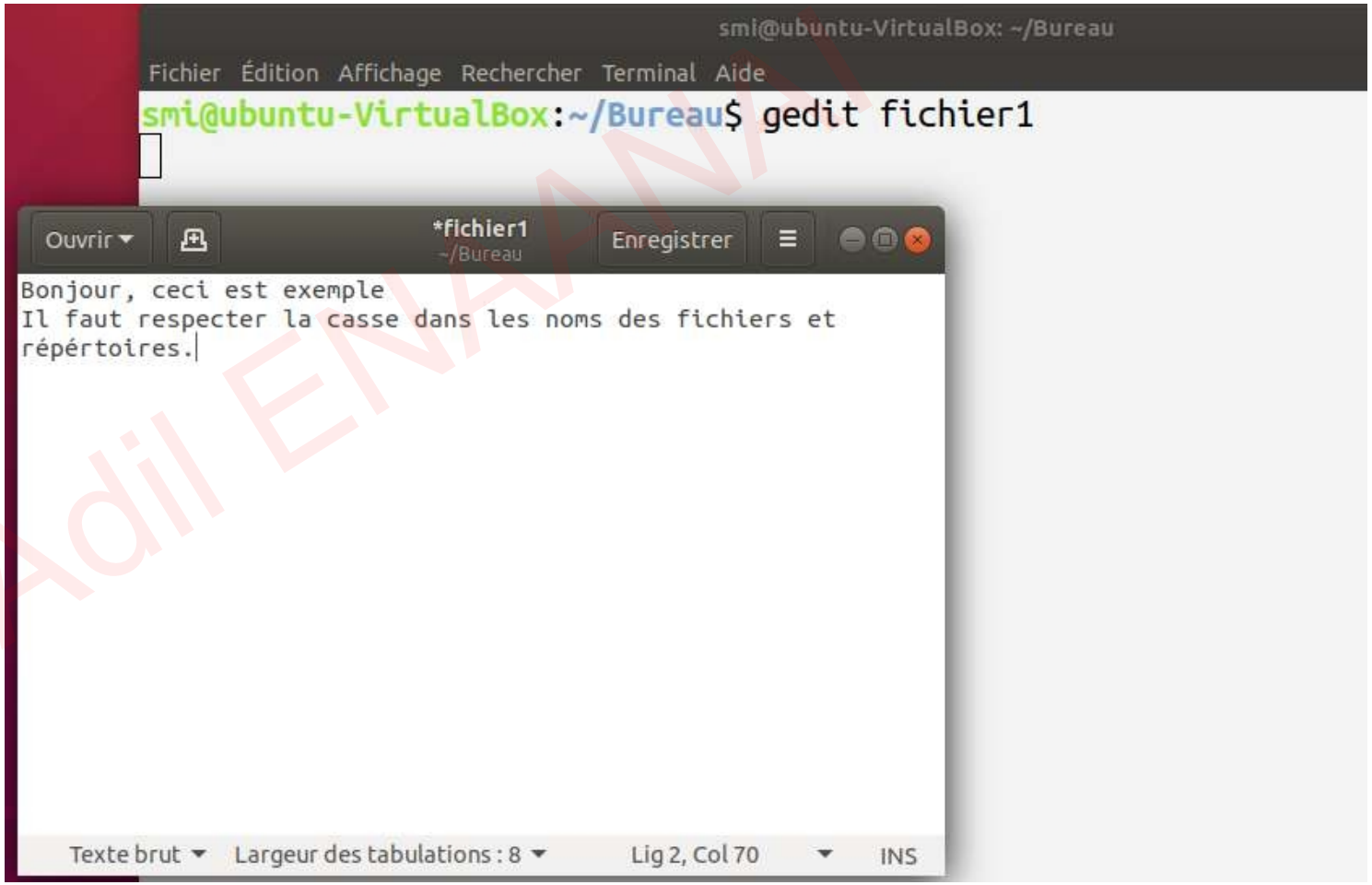
There are other command-line editors (such as Vi, Vim, nano, Emacs...) which we'll come across in the rest of this course.

If used without parameters, this allows you to edit a file as it is.

or physically created, pending registration from the editor.

File management

Example



File management View file

contents

Syntax: **cat** [*options*] [parameters]

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

Displays the contents of one or more files online. If used with the -n option, this will enumerate the file lines.

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

File management Delete a file

Syntax: **rm** [*options*] parameters

(Eng - remove files or directories)

Allows you to delete one or more files. By default, it does not delete directories. But :

This is possible with the *-r* option.

It can delete an entire tree using the option.

Dangerous?!?! Without confirmation, definitive deletion. The *-i* option executes the task interactively; requires confirmation of each operation.

File management

Example

```
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$
```

File management

Copying a file

Syntax: **cp** [*options*] parameters

(Eng - copy files and directories)

Copies a source to a destination. It can be used to copy files and directories. Examples of use. Copy a file to another file:

cp fich1 fich2

Copying files to a directory :

cp fich1 fich2 **rep_fich**

File management

Example

```
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$
```

File management Moving

(renaming) a file

Syntax: **mv** [*options*] parameters

(Eng - move (rename) files)

Moves a source to a destination directory. It can be used to move files and directories.

It can also be used to rename a file.

Examples of use:

```
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$
```

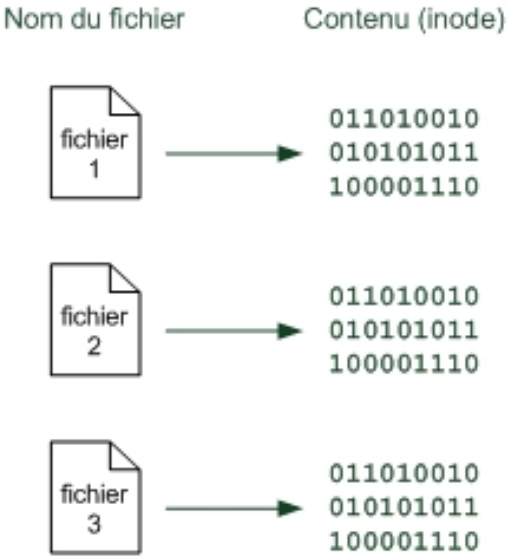
File management

Creating a physical link to a file

Creates a file in the personal folder :

touch file1

Displaying these extended properties (-i to show the inode of this file): **ls** *-li* file1 We see that see that this file a for ***inode*** 395217, and has only one link.



You can create a new link to this file, possibly in a different directory if this directory is on the same partition as the file **file1**, using the **ln** (link) command:

ln file1 Documents/file_link1

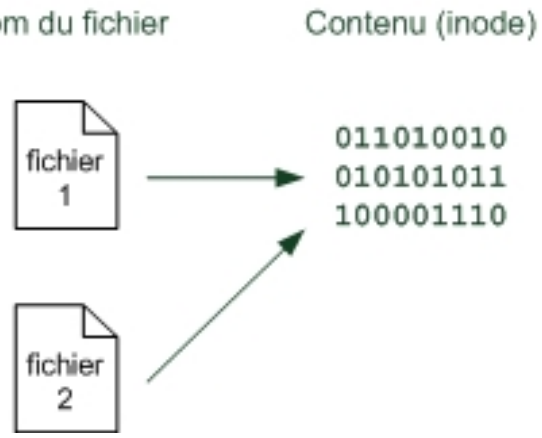
File management

Example

```
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$
```

File management

Creating a symbolic link to a file When you want to link a file via a link that is not on the same partition as the file itself, you need to use a so-called symbolic link.



ln *-s* file1 Document/SymbolicLink_file1

```
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
```

File management

Creating a symbolic link to a file

Most operations on the symbolic link are performed on the file to which it points. On the other hand, deleting this link obeys the following rules:

- The rm command deletes the symbolic link itself (which is a file at in its own right) and has no influence on the file to which it refers.
- Therefore, if you delete the file, the symbolic link still exists and points to a file that doesn't exist.
- The size of the link does not depend on the size of the file pointed to, but rather on the size corresponding to the number of characters in the path of the file to which it refers.
- The final feature of the symbolic link (which distinguishes it from the physical link) is that it is possible to create such a link on a directory.

File management

Read the first n lines of a file

Syntax: **head** *-n* file

Displays the first n lines of a file **Examples** : **head** *-5* /home/smi/Bureau/fichier1 Displays the first five lines of the file "fichier1".

Read the last n lines of a file

Syntax: **tail** *-n* file

Displays the last n lines of a file **Examples** :**tail** *-4* /home/smi/Bureau/fichier1 Displays the first four lines of the file "fichier1".

File management

Search for a string

Syntax: **grep** [*options*] *string_to_search* files Searches for a string of characters in a file.

To display the line number of the string (option -n)

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

To be case-insensitive (-i option)

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

File management

Search for a string

To display the file containing the string (-H option)

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

To display only the string file (-l option)

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

To display the number of times a string occurs (option -c)

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

File management

Search for a string

To recursively search all files in a folder

```
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
```

To display lines that do not contain the string(option -v)

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

File management Sort a

data file

Syntax: **sort** [*options*] file

Or the "*names.txt*" file on the right:

```
GNU nano 2.9.3

Mohamed
Karim
Salma
Ikram
Saad
Laila
Brahim
```

To sort names

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ sort noms.txt
Brahim
Ikram
Karim
Laila
Mohamed
Saad
Salma
```

To sort the names and put the result in another file

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

File management

Sorting a data file

Or the "names.txt" file on the right:

```
GNU nano 2.9.3

Mohamed
Karim
Salma
Ikram
Saad
Laila
Brahim
```

To sort names in reverse order

```
ubuntu@ubuntu-VirtualBox:~/Bureau$ sort -r noms.txt
Salma
Saad
Mohamed
Laila
Karim
Ikram
Brahim
```

To sort names randomly

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

File management Sort a

data file

Or the "names.txt" file on the right:

```
GNU nano 2.9.3

24
12
5
34
57
90
76
```

To sort numbers

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

12
24
34
5
57
76
90
```


It's not true

File management Sort a

data file

Or the "names.txt" file on the right:

To sort numbers with the -n option

```
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
```

Now yes

Managing Count files (

WC command)

Syntax: wc [*options*] file

Or the "names.txt" file on the right:

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

```
 GNU nano 2.9.3

Mohamed
Karim
Salma
Ikram
Saad
Laila
Brahim
```

These three numbers mean, in order :

1. the number of lines (here 7).
2. the number of words (here 7).
3. number of bytes (here 44)

NB: for the number of bytes, don't forget to count the bytes of the character "return to line

Managing Count files (

WC command)

Or the "names.txt" file on the right:

```
GNU nano 2.9.3

Mohamed
Karim
Salma
Ikram
Saad
Laila
Brahim
```

To display only the number of lines

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

To display word count only

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

To display only the number of bytes

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

File management

Remove duplicates from a file

Or the "*duplicate_names.txt*" file on the right:

To remove duplicates, use the **Uniq** command on a **sorted** file.

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

```
 GNU nano 2.9.3

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

File management

Remove duplicates from a file

Or the "*duplicate_names.txt*" file on the right:

```
  GNU nano 2.9.3

Brahim
Brahim
Ikram
Karim
Laila
```

To delete duplicates and save the result in another file

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

File management

Remove duplicates from a file

To count the number of occurrences of each name, use the **-c** option

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

File management

Remove duplicates from a file

To display duplicate names only, use the **-d** option

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

File management Cut part

of a file

Syntax: **cut** [*options*] file

Cut by number of characters

If you wish to retain only characters 2 to 5 of each line of the file, type :

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

File management Cut part

of a file

Syntax: **cut** [*options*] file

Cut by number of characters

If you wish to keep the first until the 5th of each line of the file, type :

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

If you want to keep the 5th to the last of each line in the file, type :

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

File management Cut part

of a file

Syntax: **cut** [*options*] file

Cut according to a delimiter

The "Students.txt" file on the right:

```
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
```

For first names only

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

Karim
Hasnaa
Laila
Mohamed
Rachid
```

-d: indicates the delimiter in the ;
-f: indicates the number of the field(s) to be cut.

File management Cut part

of a file

Syntax: **cut** [*options*] file

Cut according to a delimiter

The "Students.txt" file on the right:

```
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
```

For first and last names

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

To display data from the second to the fourth column

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

File management Find a

file (locate)

Syntax: **locate** [*options*] string

Locate gives you all files that contain the word "string" in their name. Whether they're files or folders, it makes no difference. It gives you the complete list of files it has found.

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

File management Find a

file (locate)

Practical example:

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

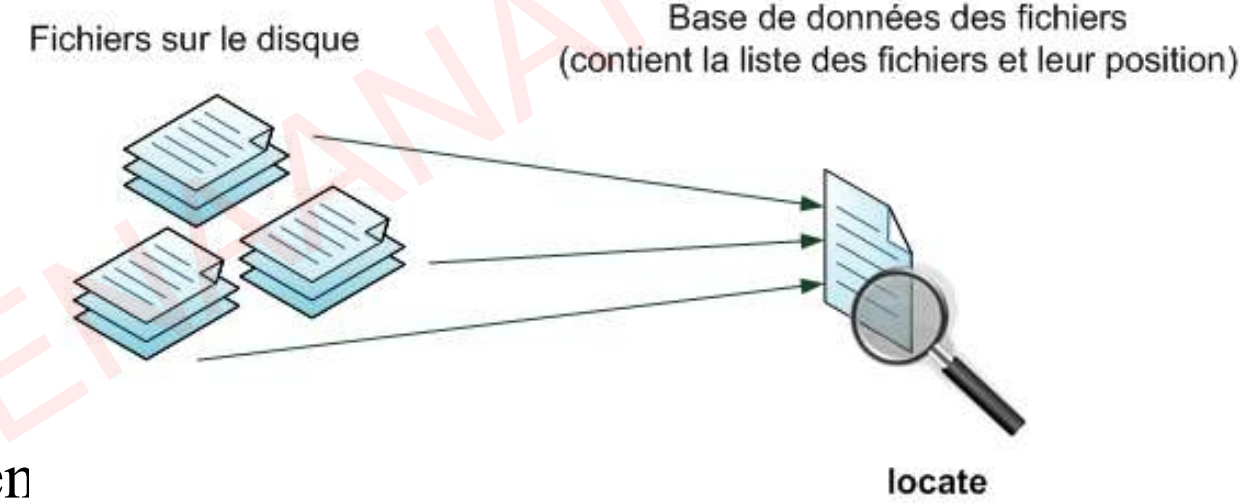
I've just created some files (using the ***touch*** command, for example), and **locate** doesn't return any results. Why is this?

the **locate** command doesn't search your entire hard disk, but only a database of your files.

File management

Find a file (locate)

Practical example:



the files have just been
database. They will therefore not be discovered by **locate**.
Once a day, your system will update the database. So, if you try again tomorrow, it's likely that **locate will** finally find your file.

File management Find a

file (locate)

Practical example:

I have to wait 24 hours to update the file database?

You can force the **locate** command to rebuild the hard disk file database. This is done with the **updatedb** command, run as **root** (with ***sudo***):

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

File management Find a

file

find is the search command par excellence for finding files, but also for performing operations on each of the files found. It's very powerful, so it can do a lot of things, and as a result... it's a bit complex.

Unlike **locate**, **find** doesn't read from a database, but instead scans your entire hard disk. This can be very time-consuming if you have several gigabytes of data!

File management Find a

file

The **find** command is used as follows:

find "where" "what" "what to do with"
Only the "what" parameter is mandatory.

Where: Name of the folder in which the command will search. What: The file to be searched. You can search for a file by name, but also by creation date or size.
What do with : it is possible to on actions automatically on each file found (known as "post-processing").

File management Find a

file

Basic use of the find command

Search by name

```
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
```

Search path

By default, find is case-sensitive, i.e. it is case-sensitive. A search for "photo.jpg" will not find the file "Photo.jpg". To make find case-insensitive, use the **-iname** parameter instead of **-name**.

File management Find a

file

Basic use of the find command

Search by size

Search for files larger than 4200Kb Search for files smaller

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

than 5Mb

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

Search for files with a size between 4MB and 5MB If the size is

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

followed by c, the value is expressed in bytes.

File management Find a

file

Basic use of the find command

Search by last access date

With **-atime**, you can specify the number of days you'd like to separates the last access to a file.

```
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
```

Strictly less than one day

Files beginning with 'names

Greater than one day

File management Find a

file

Basic use of the find command

Search only directories or files

You can also search only for directories or files. We use:

-type d: to search directories only;

-type f: to search for files only.

File management Find a

file

Basic use of the find command

Display files in a formatted way

By default, only the file names found are listed. However, with the -printf option, which will remind some of the C language, you can manipulate what's displayed.

```
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
```

With **%p**: file path; **%s**: file size; **%u**: user name

File management Find a

file

Basic use of the find command

Delete files found

To delete all files found

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

Calling up an order

With **-exec**, you can call a command that will perform an action on each of the files found.

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

Exercise: try grouping all the .jpg files scattered around your home directory into an images folder.