

# UNIX

## Lecture 1: Introduction to UNIX

Filipe Vasconcelos<sup>1</sup>

<sup>1</sup>ESME, Lille, [filipe.vasconcelo@esme.fr](mailto:filipe.vasconcelo@esme.fr)

- ① Syllabus
- ② Prerequisites – Session 0
- ③ Unix introduction

## ① Syllabus

- 1.1 Main Objective of the Course
- 1.2 Intended Learning Outcomes
- 1.3 Course Content
- 1.4 Assessment Methods
- 1.5 Learning Resources

## ② Prerequisites – Session 0

## ③ Unix introduction



1

# Syllabus

## ① Syllabus

- 1.1 Main Objective of the Course
- 1.2 Intended Learning Outcomes
- 1.3 Course Content
- 1.4 Assessment Methods
- 1.5 Learning Resources

## ② Prerequisites – Session 0

## ③ Unix introduction

- 3.1 Operating System
- 3.2 UNIX Operating System
- 3.3 File system
- 3.4 Basic commands



# Main Objective of the Course

At the end of this course, students will be able to interact with the `bash` command interpreter, which is widely associated with **UNIX**-like operating systems.



## ① Syllabus

- 1.1 Main Objective of the Course
- 1.2 Intended Learning Outcomes
- 1.3 Course Content
- 1.4 Assessment Methods
- 1.5 Learning Resources

## ② Prerequisites – Session 0

## ③ Unix introduction

- 3.1 Operating System
- 3.2 UNIX Operating System
- 3.3 File system
- 3.4 Basic commands



# Intended Learning Outcomes

**ILO1:** Identify or recognize an operating system from the Unix family and provide some examples of these operating systems (Linux, macOS, BSD). Install and use a Bash command interpreter, whether through an emulator, a virtual machine, or a complete Unix family operating system.

**ILO2:** Recognize, describe, and use basic Bash commands, or quickly find documentation about these commands.

**ILO3:** Use and chain together, through pipes, the main filter commands (grep, cut, head, tail, etc.) to manipulate data streams, whether or not coming from files.

**ILO4:** Produce, test, and verify the results of Shell scripts (in Bash) to perform tasks ranging from simple to complex algorithms.

**ILO5:** Analyze and explore the file system hierarchy of a Unix-type OS using basic commands, manage the access rights of directories and files within the hierarchy and manage process.

**ILO6:** Design, comment, and justify the command-line interface of a developed script.



## ① Syllabus

- 1.1 Main Objective of the Course
- 1.2 Intended Learning Outcomes
- 1.3 Course Content
- 1.4 Assessment Methods
- 1.5 Learning Resources

## ② Prerequisites – Session 0

## ③ Unix introduction

- 3.1 Operating System
- 3.2 UNIX Operating System
- 3.3 File system
- 3.4 Basic commands



## 11 Lecture Sessions (2 hours):

- ❑ Introduction to UNIX (Absolute/Relative Paths)
- ❑ Basic Commands, Shell Metacharacters
- ❑ grep Commands and Regular Expressions
- ❑ Input/Output Redirection and Pipes
- ❑ User Management and Access Rights
- ❑ Process Management
- ❑ SHELL Programming (bash) (Part 1)
- ❑ SHELL Programming (bash) (Part 2)
- ❑ SHELL Programming (bash) (Part 3)
- ❑ SHELL Programming (bash) (Part 4)
- ❑ Filtering Commands
- ❑ Review Session – Final Exam Preparation



## ① Syllabus

- 1.1 Main Objective of the Course
- 1.2 Intended Learning Outcomes
- 1.3 Course Content
- 1.4 Assessment Methods**
- 1.5 Learning Resources

## ② Prerequisites – Session 0

## ③ Unix introduction

- 3.1 Operating System
- 3.2 UNIX Operating System
- 3.3 File system
- 3.4 Basic commands



$$\text{TOTAL} = 0.5\text{CC} + 0.5\text{EX}$$

- ❑ CC : Continuous assessment
  - ❑ Weekly Moodle quizzes at the beginning of the session ( 10 multiple-choice questions).
  - ❑ Questions cover the two previous sessions.
  - ❑ The first quiz (session 2) is formative.
  - ❑ Two jokers, average of the best 8 scores. (An absence = 0).
- ❑ EX : Final exam
  - ❑ Exam on Moodle + SEB.
  - ❑ Bash scripting.
  - ❑ The last course session is a revision session using past exams.



## ① Syllabus

- 1.1 Main Objective of the Course
- 1.2 Intended Learning Outcomes
- 1.3 Course Content
- 1.4 Assessment Methods
- 1.5 Learning Resources

## ② Prerequisites – Session 0

## ③ Unix introduction

- 3.1 Operating System
- 3.2 UNIX Operating System
- 3.3 File system
- 3.4 Basic commands



## Moodle:

- Course PDFs, exercises and solutions for tutorials (may be updated during the year).
- Weekly quizzes
- Video resources to complement the lectures
- Past exam papers

## Student GitHub:

- PDFs (updated at the beginning of the year)
- Shared files for in-class activities



- ① Syllabus**
- ② Prerequisites – Session 0**
- ③ Unix introduction**





2

## Prerequisites – Session 0



Have you installed a bash command interpreter?  
(WSL, VM or a Linux system . . . ?)



- ① Syllabus
- ② Prerequisites – Session 0
- ③ Unix introduction
  - 3.1 Operating System
  - 3.2 UNIX Operating System
  - 3.3 File system
  - 3.4 Basic commands





3

# Unix introduction



## ILO1

Identify or recognize an operating system from the Unix family and provide some examples of these operating systems (Linux, macOS, BSD). Install and use a Bash command interpreter, whether through an emulator, a virtual machine, or a complete Unix family operating system.



## ① Syllabus

- 1.1 Main Objective of the Course
- 1.2 Intended Learning Outcomes
- 1.3 Course Content
- 1.4 Assessment Methods
- 1.5 Learning Resources

## ② Prerequisites – Session 0

## ③ Unix introduction

- 3.1 Operating System
- 3.2 UNIX Operating System
- 3.3 File system
- 3.4 Basic commands





**Launch the Wooclap activity of the current lecture:  
Lecture01**

*What words come to mind when you hear the term "operating system"?*



- ❑ Examples of operating systems:

- ❑ Desktop: Windows 11/10/8/7, MacOS, UNIX, Linux
  - ❑ Smartphone: iOS, Android

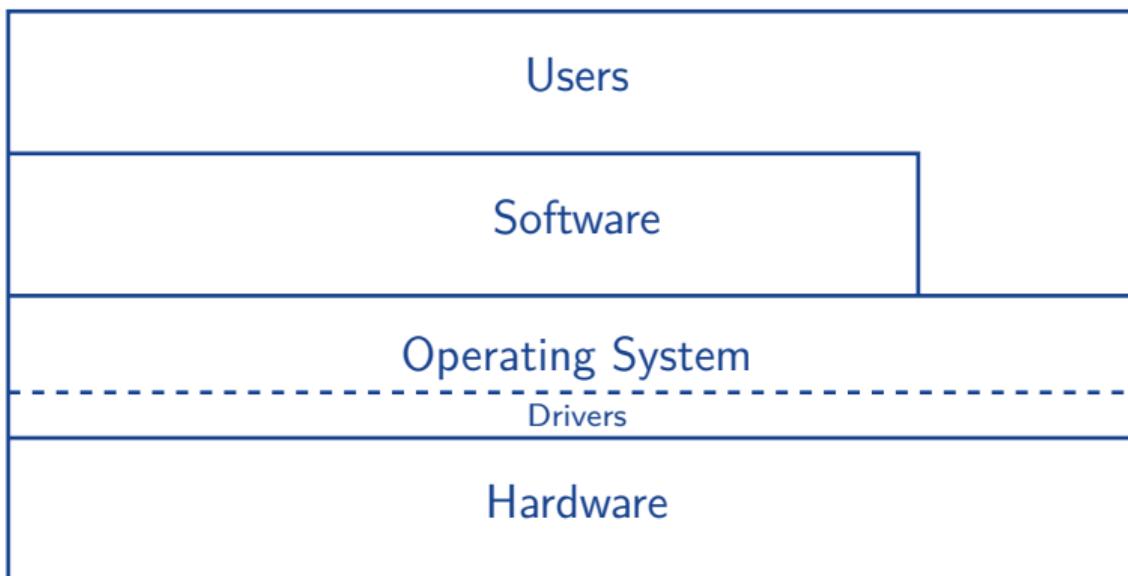
- ❑ Examples of everyday use:

- ❑ Web browsing: (It controls the network card, enables data transmission via the internet, etc.)
  - ❑ Editing a file: (It manages the computer's file system)



# Operating System

- ❑ An operating system is an interface between the user and hardware layers of a computer. It is a computer program launched at startup, enabling optimal management of a computer's physical resources.



## ① Syllabus

- 1.1 Main Objective of the Course
- 1.2 Intended Learning Outcomes
- 1.3 Course Content
- 1.4 Assessment Methods
- 1.5 Learning Resources

## ② Prerequisites – Session 0

## ③ Unix introduction

- 3.1 Operating System
- 3.2 UNIX Operating System
- 3.3 File system
- 3.4 Basic commands



# Your feedback



**Launch the Wooclap activity of the current lecture:  
Lecture01**

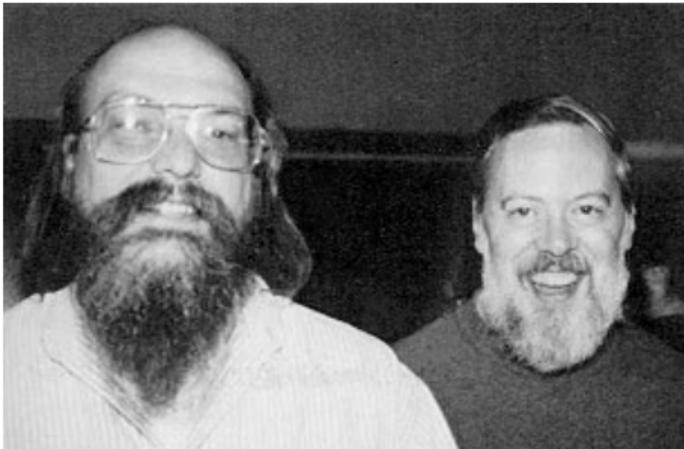
*What words come to mind when you hear "Unix"?*



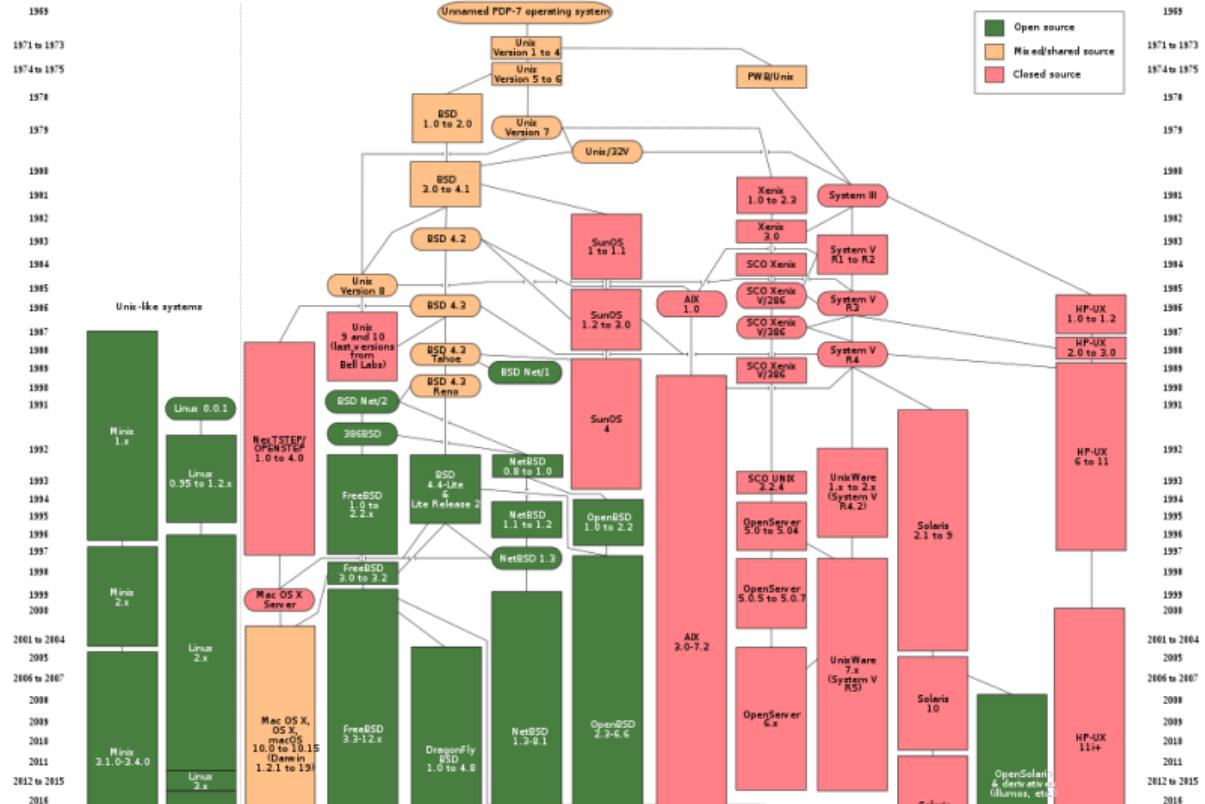
# History of UNIX

The first version of UNIX was developed in 1969 by Ken Thompson and Dennis Ritchie at Bell Labs, part of AT&T.

- ❑ The company later distributed UNIX to students and professors at universities.
- ❑ UNIX was distributed with its source code at that time, which allowed for many independent developments and contributions.



# The big UNIX family tree

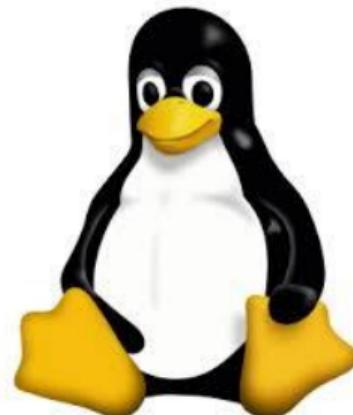


# The GNU Project

- ❑ Richard Stallman created the GNU project in 1984.
- ❑ GNU is a UNIX-like operating system. This means it consists of many programs: applications, libraries, development tools.
- ❑ The name *GNU* is a recursive acronym for *GNU's Not UNIX*.
- ❑ GNU programs are distributed under the GPL free software license.



- ❑ Developed at the University of Helsinki in 1991 by Linus Torvalds.
- ❑ Linux is a kernel of an UNIX-like operating system.
- ❑ Linux is open-source software.



- ❑ These two projects are complementary: while Richard Stallman created the basic programs (file copy, file deletion, text editor), Linus worked on creating the "heart" of an operating system: the kernel.
- ❑ The GNU project (free software) and Linux (OS kernel) merged to create GNU/Linux.
- ❑ It is GNU/Linux that is generally simply called Linux.



## Linux Usage

- ❑ Server (Apache Web server, SQL database server, PHP server, etc.)
- ❑ Network security (gateway, router, proxy, firewall, etc.)
- ❑ Mainframe (computing centers in banks, insurance companies, etc.)
- ❑ Server cluster (clusters of Google, Amazon, Microsoft, etc.)
- ❑ Supercomputers (high-performance computing)
- ❑ Embedded system (internet boxes, smartphones, digital TV decoders, PlayStation, Raspberry Pi, etc.)



# GNU/Linux Distribution

- ❑ Linux itself is just a kernel, and it needs to be accompanied by other software to form a complete operating system.
- ❑ Several companies or associations distribute GNU/Linux along with a coherent set of software: what is called a Linux distribution.



# Linux Distribution

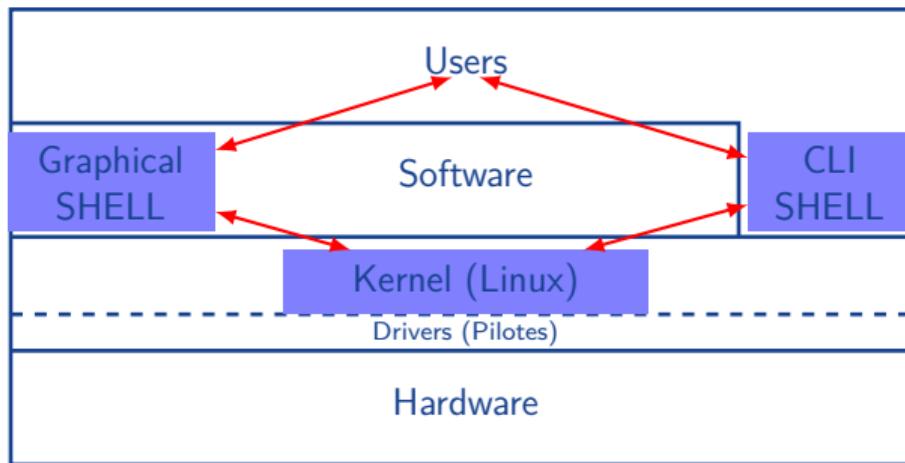


debian



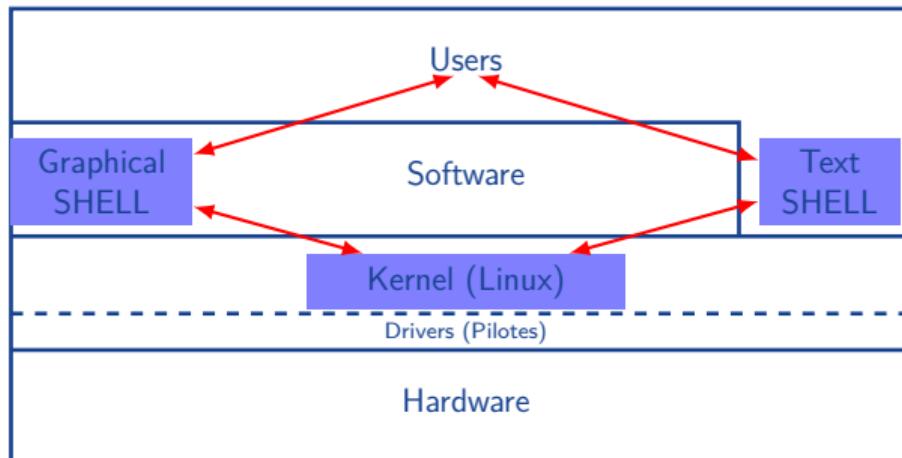
## GUI: Graphical User Interface

- ❑ There are several graphical environments on GNU/Linux:
  - ❑ GTK: Gnome, Unity, MATE, Xfce, etc.
  - ❑ Qt: KDE, LXQt, etc.
  - ❑ WSL (Windows): Xming



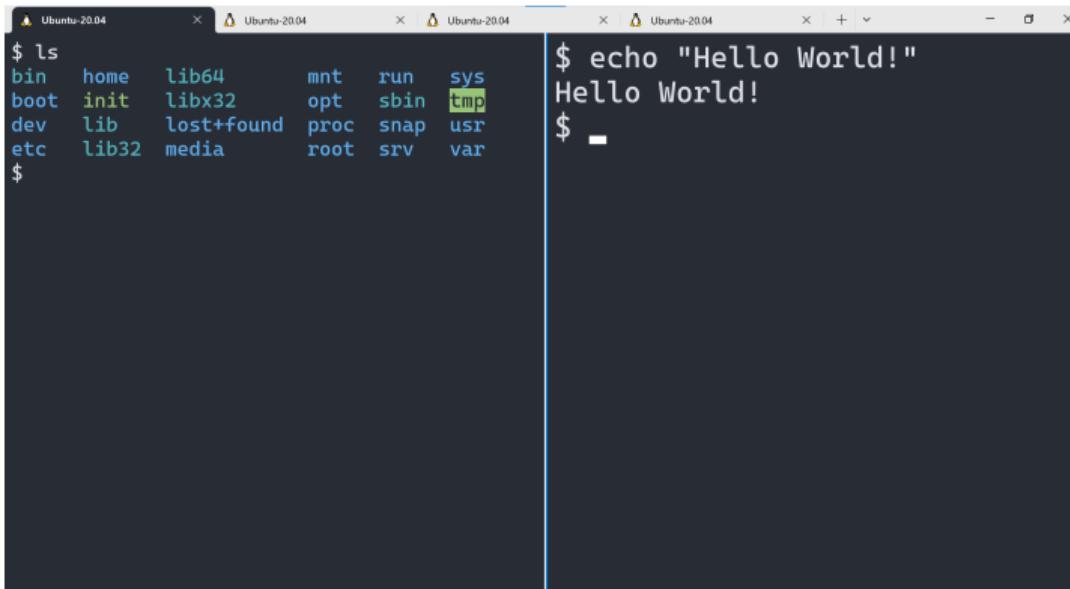
## CLI: *Command Line Interface*

- ❑ There are different versions of command interpreters:
  - ❑ **bash: Bourne Again Shell**
  - ❑ sh: Bourne Shell
  - ❑ zsh: Z Shell



# Command Line Interface (CLI)

- ❑ More efficient and faster for issuing commands to the computer.
- ❑ Consumes fewer hardware resources to operate.
- ❑ Allows automation of a sequence of instructions to be executed via scripts (next lesson).



The image shows three separate terminal windows side-by-side, all titled "Ubuntu-20.04".

- The leftmost window displays the command `$ ls` followed by a list of system directories: bin, home, lib64, mnt, run, sys, boot, init, libx32, opt, sbin, tmp, dev, lib, lost+found, proc, snap, usr, etc, lib32, media, root, srv, var, and a final dollar sign.
- The middle window displays the command `$ echo "Hello World!"` followed by the output "Hello World!" and a final dollar sign.
- The rightmost window is mostly blank, showing only a single dollar sign.

## ① Syllabus

- 1.1 Main Objective of the Course
- 1.2 Intended Learning Outcomes
- 1.3 Course Content
- 1.4 Assessment Methods
- 1.5 Learning Resources

## ② Prerequisites – Session 0

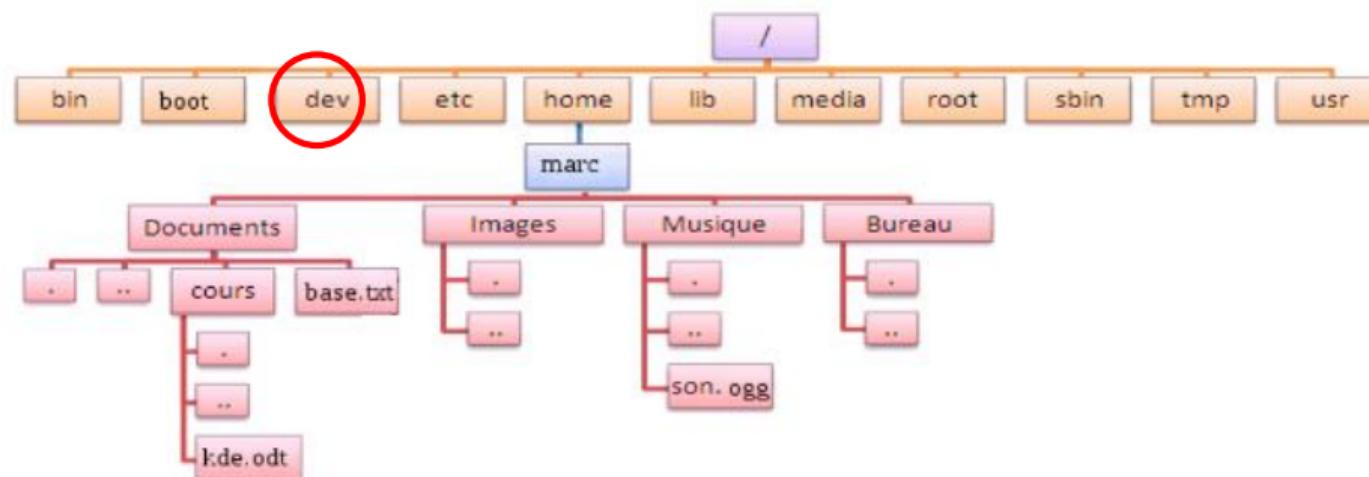
## ③ Unix introduction

- 3.1 Operating System
- 3.2 UNIX Operating System
- 3.3 File system
- 3.4 Basic commands



# Everything is file

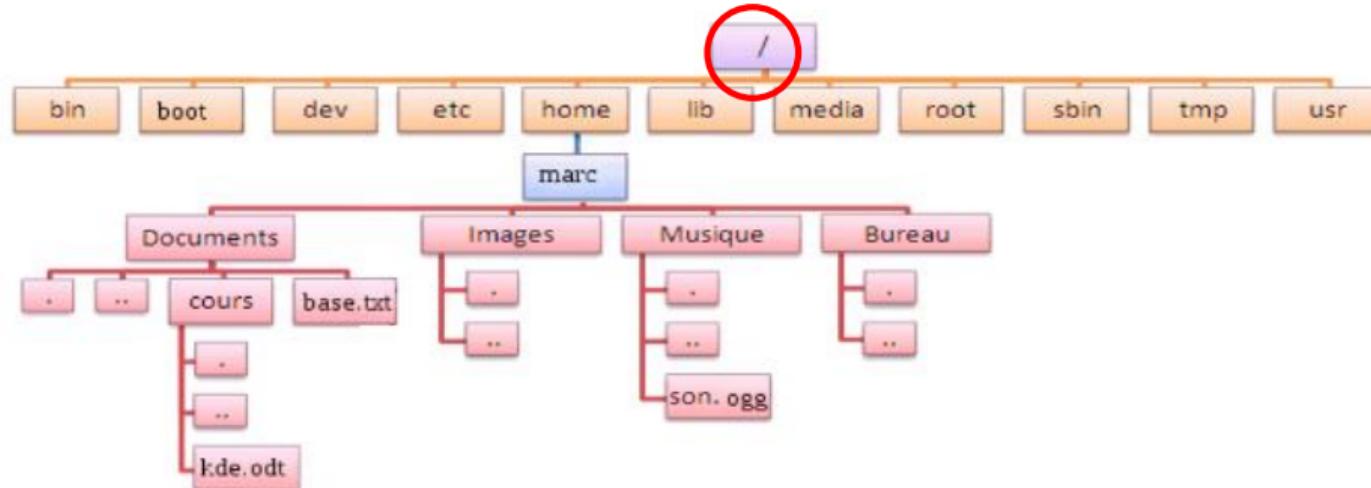
- ❑ Disk, USB drive, processor, networks, graphics card, sound card, etc.
- ❑ The hardware organization in Linux is very different from that in Windows.
- ❑ Example: In Linux, plugging in a device is akin to creating a new file in '/dev'.



# Directory Structure (Tree)

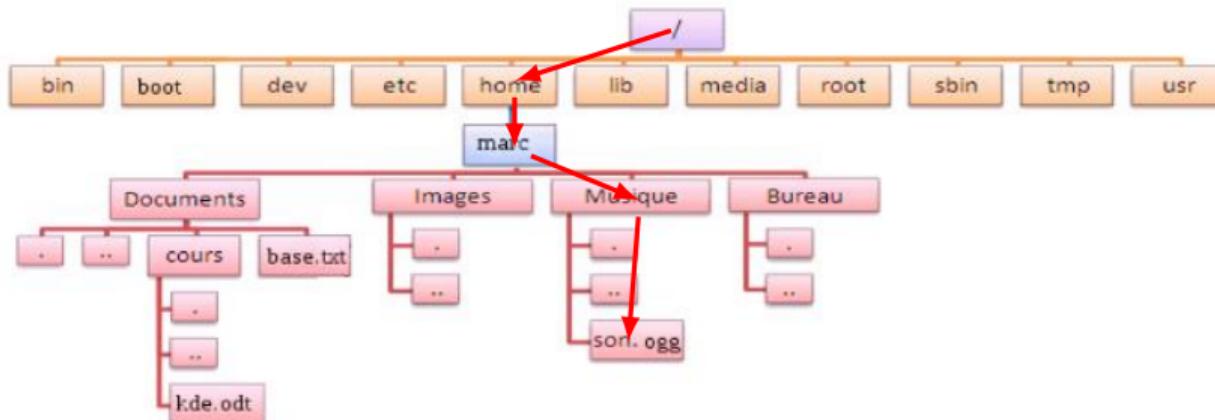
Directories are nodes in a tree

- ❑ The root directory is at the top of the hierarchy.
- ❑ The system knows the root.
- ❑ In Linux, the root is denoted by '/': the starting point of the tree.



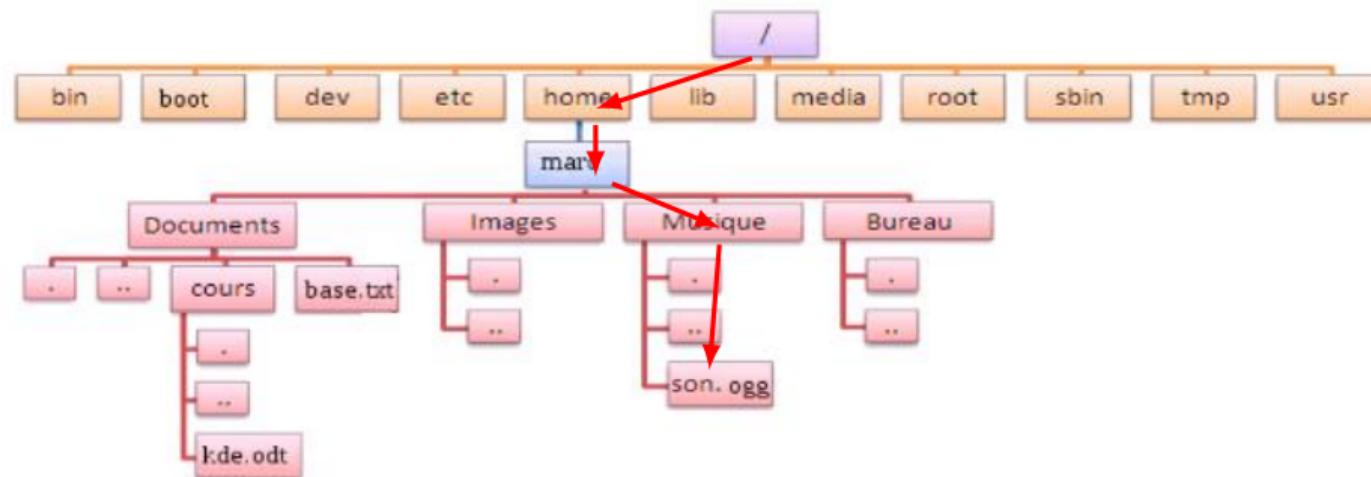
How to locate a file in this directory tree?

- ❑ A file name consists of two parts: the path (e.g., directory names to navigate through separated by /) and the file name itself.
- ❑ Example: /home/marc/Music/song.ogg



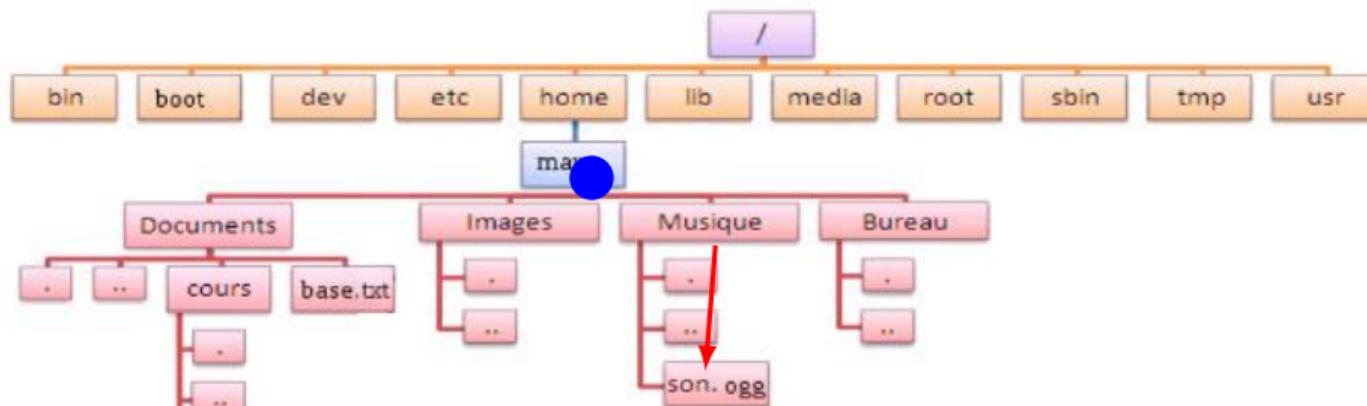
## Absolute Path

- ❑ Starting directory: Root directory
- ❑ An absolute path starts with '/'
- ❑ Example: /home/marc/Music/song.ogg



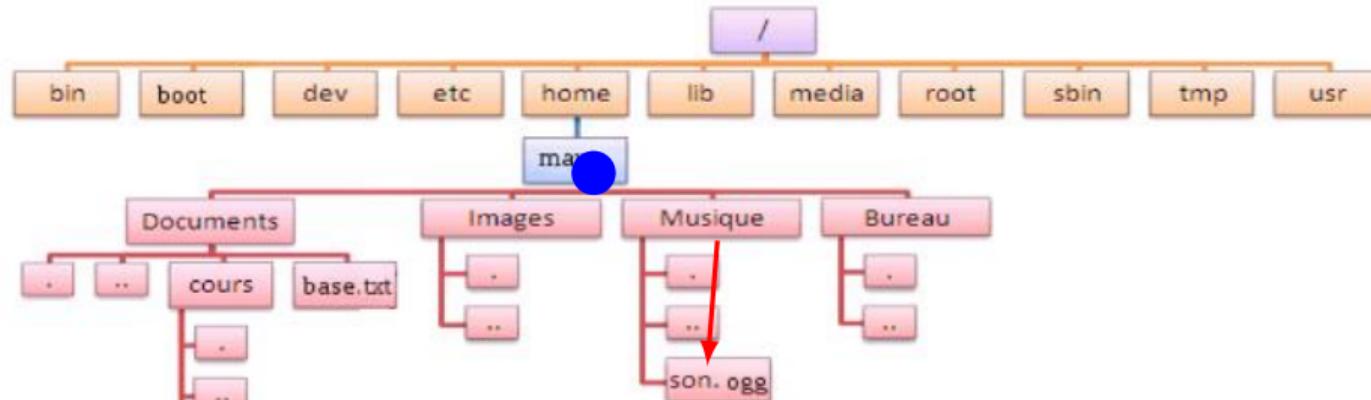
## Relative Path

- ❑ Starting directory: Current directory
- ❑ A relative path starts with a file or directory name
- ❑ Example: If /home/marc is the current directory, Music/song.ogg is the relative path to song.ogg.



## Relative Path (Continued)

- ❑ Relative name of the current directory: .(dot)
- ❑ Example: ./Music/song.ogg is equivalent to Music/song.ogg
- ❑ Relative name of the parent of the current directory: .. (dot-dot)
- ❑ Example: ../marc/Music/song.ogg is equivalent to Music/song.ogg



## What is a Directory?

- A directory is a list of files it contains.
- A directory is never empty:
- It contains at least information about itself: . (dot) and its parent: .. (dot-dot)
- Example: If the current directory is /home/marc/Music, you can access the file in the Documents directory using the relative path ../Documents.



# Absolute vs. Relative Paths

## Absolute Paths

- Usable in any context
- Practical in scripts: Independent of the current directory
- Convenient for specifying a file or directory that is not in the immediate vicinity of the current directory. You don't have to count the  
.../.../.../.../.

## Relative Paths

- Short names for finding files or directories in the immediate vicinity



## Directories in UNIX-like Systems

- ❑ / : Root directory
- ❑ /root : Personal directory of the root user (administrator)
- ❑ /home : User directories
- ❑ /Users: User directories (MacOS)
- ❑ /usr : Important directory in the Linux system. It contains data and programs shared among users.



## ① Syllabus

- 1.1 Main Objective of the Course
- 1.2 Intended Learning Outcomes
- 1.3 Course Content
- 1.4 Assessment Methods
- 1.5 Learning Resources

## ② Prerequisites – Session 0

## ③ Unix introduction

- 3.1 Operating System
- 3.2 UNIX Operating System
- 3.3 File system
- 3.4 Basic commands





File Commands	System Info
<code>ls</code> - directory listing	
<code>ls -al</code> - formatted listing with hidden files	
<code>cd dir</code> - change directory to <code>dir</code>	
<code>cd ..</code> - change to home	
<code>pwd</code> - show current directory	
<code>mkdir dir</code> - create a directory <code>dir</code>	
<code>rm file</code> - delete <code>file</code>	
<code>rm -r dir</code> - delete directory <code>dir</code>	
<code>rm -f file</code> - force remove <code>file</code>	
<code>rm -rf dir</code> - force remove directory <code>dir</code> *	
<code>cp file1 file2</code> - copy <code>file1</code> to <code>file2</code>	
<code>cp -r dir1 dir2</code> - copy <code>dir1</code> to <code>dir2</code> ; create <code>dir2</code> if it doesn't exist	
<code>mv file1 file2</code> - rename or move <code>file1</code> to <code>file2</code> if <code>file2</code> is an existing directory, moves <code>file1</code> into directory <code>file2</code>	
<code>ln -s file link</code> - create symbolic link <code>link</code> to <code>file</code>	
<code>touch file</code> - create or update <code>file</code>	
<code>cat &gt; file</code> - places standard input into <code>file</code>	
<code>more file</code> - output the contents of <code>file</code>	
<code>head file</code> - output the first 10 lines of <code>file</code>	
<code>tail file</code> - output the last 10 lines of <code>file</code>	
<code>tail -f file</code> - output the contents of <code>file</code> as it grows, starting with the last 10 lines	
Process Management	Compression
<code>ps</code> - display your currently active processes	<code>tar cf file.tar files</code> - create a tar named <code>file.tar</code> containing <code>files</code>
<code>top</code> - display all running processes	<code>tar xf file.tar</code> - extract the files from <code>file.tar</code>
<code>kill pid</code> - kill process id <code>pid</code>	<code>tar czf file.tar.gz files</code> - create a tar with Gzip compression
<code>killall proc</code> - kill all processes named <code>proc</code> *	<code>tar xzf file.tar.gz</code> - extract a tar using Gzip
<code>bg</code> - lists stopped or background jobs; resume a stopped job in the background	<code>tar cjf file.tar.bz2</code> - create a tar with Bzip2 compression
<code>fg</code> - brings the most recent job to foreground	<code>tar xjf file.tar.bz2</code> - extract a tar using Bzip2
<code>fg n</code> - brings job <code>n</code> to the foreground	<code>gzip file</code> - compresses <code>file</code> and renames it to <code>file.gz</code>
	<code>gzip -d file.gz</code> - decompresses <code>file.gz</code> back to <code>file</code>
File Permissions	Network
<code>chmod octal file</code> - change the permissions of <code>file</code> to <code>octal</code> , which can be found separately for user, group, and world by adding:	<code>ping host</code> - ping <code>host</code> and output results
<ul style="list-style-type: none"><li>• 4 - read (r)</li><li>• 2 - write (w)</li><li>• 1 - execute (x)</li></ul>	<code>whois domain</code> - get whois information for <code>domain</code>
	<code>dig domain</code> - get DNS information for <code>domain</code>
	<code>dig -x host</code> - reverse lookup <code>host</code>
	<code>wget file</code> - download <code>file</code>
	<code>wget -c file</code> - continue a stopped download
Installation	
	Install from source: <code>./configure</code>

## Command

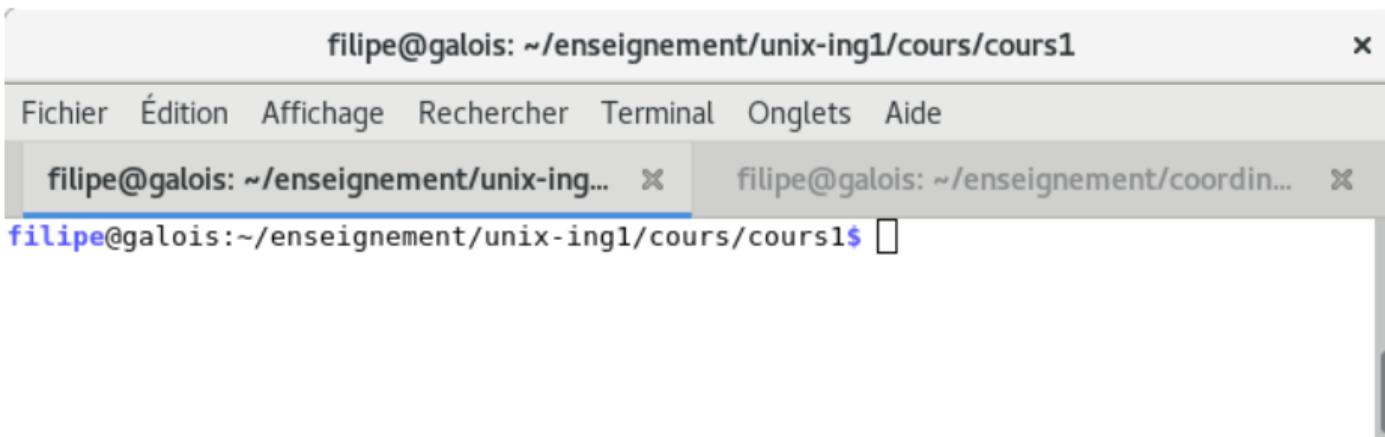
- ❑ A command contains separators that allow you to isolate commands, options, and arguments (the space character is one of these separators).
- ❑ The first element of the command consists of either:
  - ❑ The name of an internal command in Linux.
  - ❑ The name of an executable (e.g., a script).



# First Commands

A command line is typed in front of a prompt

- ❑ prompt: A string of characters indicating that the interpreter is waiting for a new command.

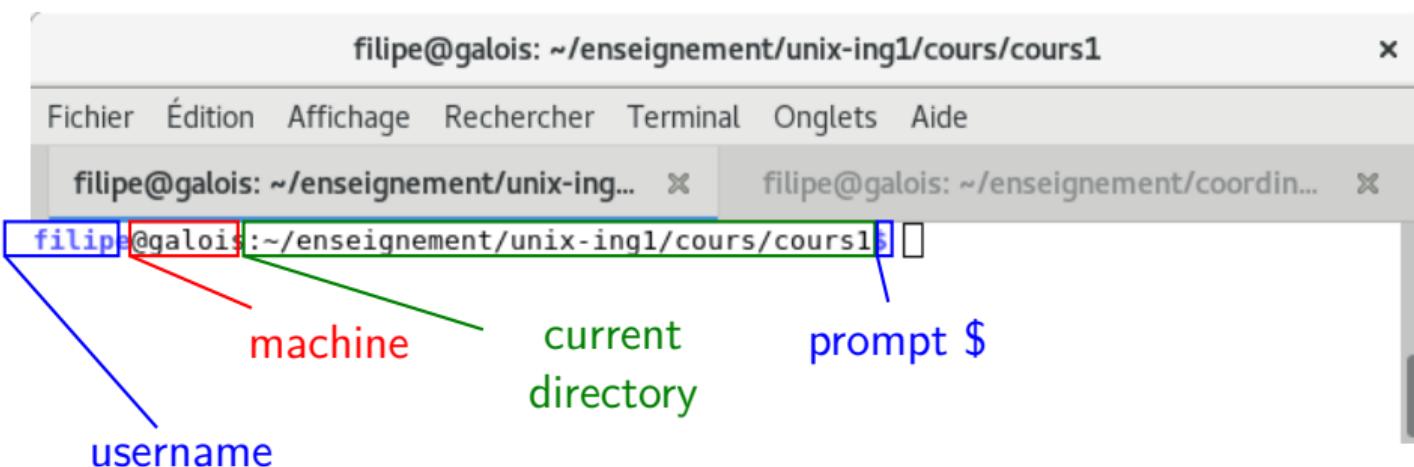


The screenshot shows a terminal window with a light gray background. At the top, there is a blue header bar with the text "filipe@galois: ~/enseignement/unix-ing1/cours/cours1" and a close button (an 'x'). Below the header, there is a menu bar with French labels: "Fichier", "Édition", "Affichage", "Rechercher", "Terminal", "Onglets", and "Aide". The main area of the terminal contains two tabs: "filipe@galois: ~/enseignement/unix-ing..." and "filipe@galois: ~/enseignement/coordin...". The bottom tab is currently active, showing the command line "filipe@galois:~/enseignement/unix-ing1/cours/cours1\$". To the right of the command line is a small black square icon with a white diagonal line through it, which typically represents a copy or paste function. The terminal window has a vertical scroll bar on the right side.

# First Commands

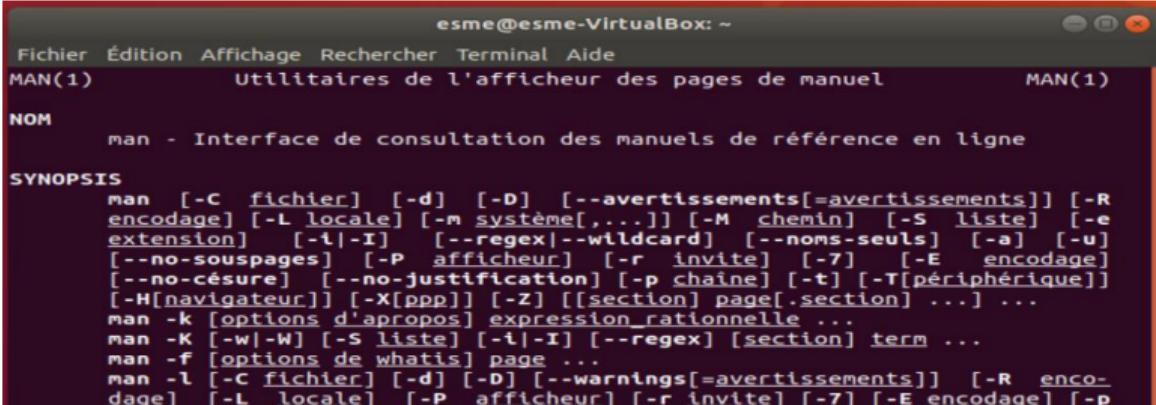
A command line is typed in front of a prompt

- ❑ prompt: A string of characters indicating that the interpreter is waiting for a new command.



## man manual

- ❑ Usage: Command to access the manual pages installed on the system.
- ❑ Syntax: `man [-k] [section]` command -k: Searches for short descriptions of manual pages that match the keyword and displays all found descriptions.
- ❑ Example: `man man`



The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "esme@esme-VirtualBox: ~". The menu bar includes "Fichier", "Édition", "Affichage", "Rechercher", "Terminal", and "Aide". The main area displays the man page for "man".

```
Fichier Édition Affichage Rechercher Terminal Aide
MAN(1)           Utilitaires de l'afficheur des pages de manuel          MAN(1)

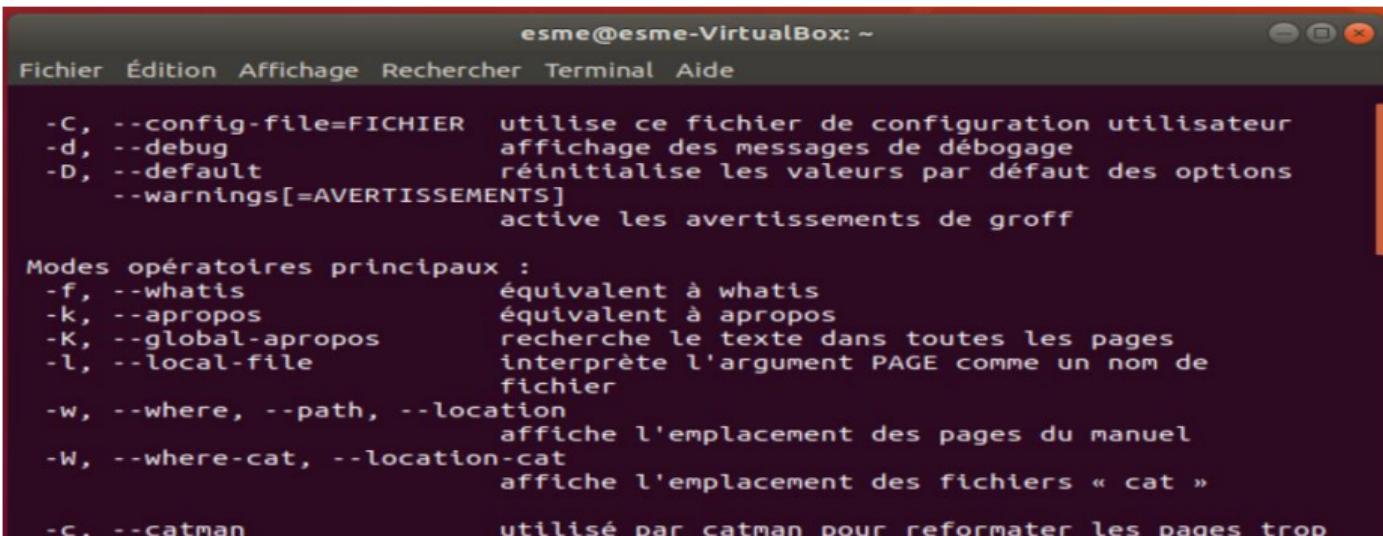
NOM
    man - Interface de consultation des manuels de référence en ligne

SYNOPSIS
    man [-C fichier] [-d] [-D] [--avertissements[=avertissements]] [-R
    encodage] [-L locale] [-m système[,...]] [-M chemin] [-S liste] [-e
    extension] [-t|-I] [--regex|-wildcard] [--noms-seuls] [-a] [-u]
    [--no-souspages] [-P afficheur] [-r invite] [-7] [-E encodage]
    [--no-césure] [--no-justification] [-p chaîne] [-t] [-T[périphérique]]
    [-H[navigateur]] [-X[ppp]] [-Z] [[section] page|.section] ...
    man -k [options d'apropos] expression_rationnelle ...
    man -K [-w|-W] [-S liste] [-t|-I] [--regex] [section] term ...
    man -f [options de whatis] page ...
    man -l [-C fichier] [-d] [-D] [--warnings[=avertissements]] [-R enco-
    dage] [-L locale] [-P afficheur] [-r invite] [-7] [-E encodage] [-p
```

# Option -help

## -help

- ❑ Usage: You can also use the -help or -h option with a command to get help.
- ❑ Example: man -help



The screenshot shows a terminal window titled "esme@esme-VirtualBox: ~". The menu bar includes "Fichier", "Édition", "Affichage", "Rechercher", "Terminal", and "Aide". The terminal displays the help documentation for the "man" command in French. It lists various options and their descriptions, such as "-C" for configuration file, "-d" for debug mode, and "-f" for whatis mode. It also describes modes like "global-apropos" and "local-file". The text is in French, with some English terms like "PAGE" and "cat" appearing in the original command-line interface.

```
esme@esme-VirtualBox: ~
Fichier Édition Affichage Rechercher Terminal Aide

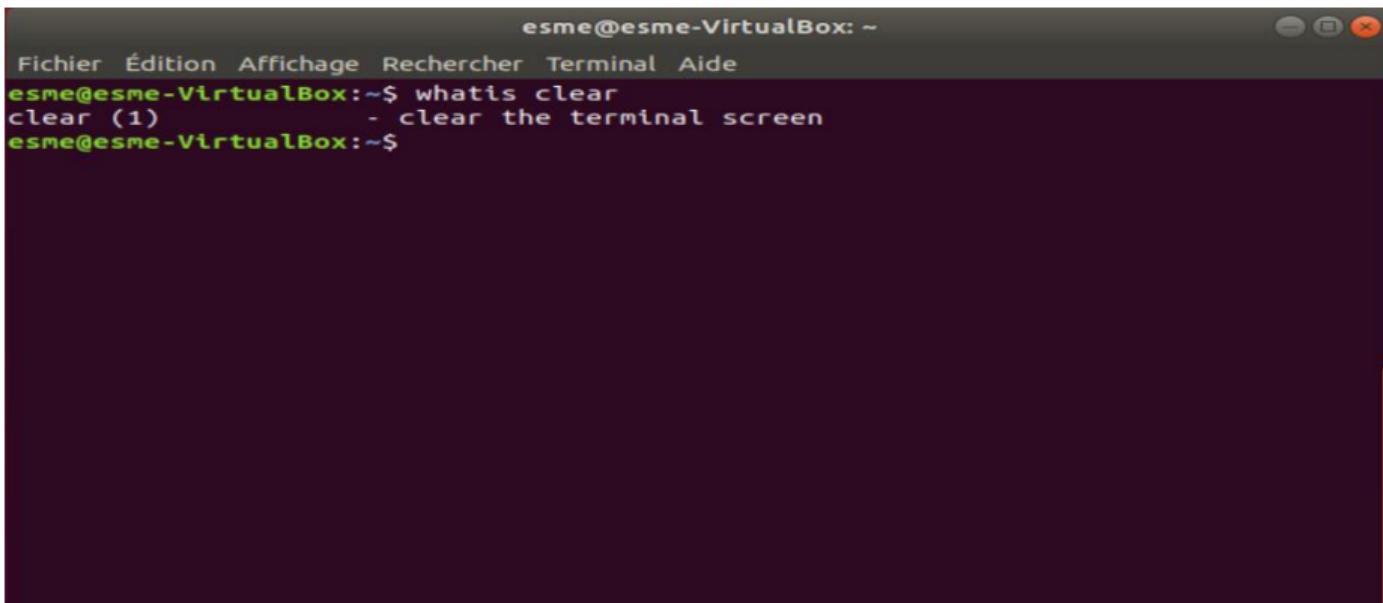
-C, --config-file=FICHIER utilise ce fichier de configuration utilisateur
-d, --debug               affichage des messages de débogage
-D, --default              réinitialise les valeurs par défaut des options
--warnings[=AVERTISSEMENTS] active les avertissements de groff

Modes opératoires principaux :
-f, --whatis            équivalent à whatis
-k, --apropos           équivalent à apropos
-K, --global-apropos    recherche le texte dans toutes les pages
-l, --local-file         interprète l'argument PAGE comme un nom de
                        fichier
-w, --where, --path, --location
                        affiche l'emplacement des pages du manuel
-W, --where-cat, --location-cat
                        affiche l'emplacement des fichiers « cat »
-c, --catman
                        utilisé par catman pour reformater les pages trop
```

# Command whatis

## whatis

- ❑ Usage: Display a one-line description of manual pages.
- ❑ Example: whatis clear



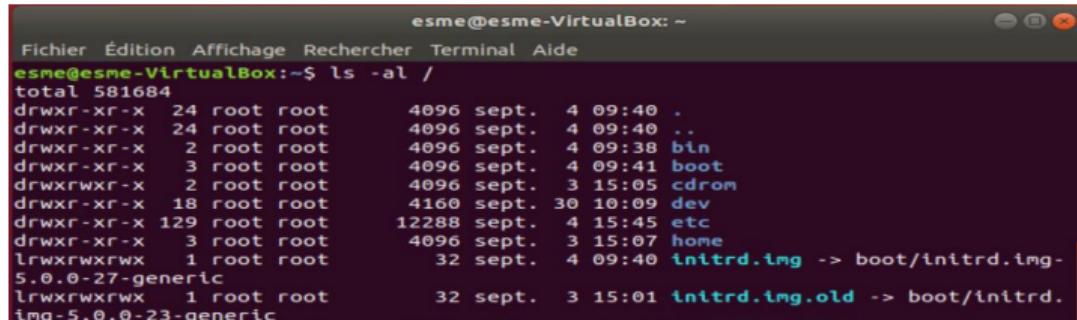
The screenshot shows a terminal window with a dark background and light-colored text. At the top, there is a menu bar with options: Fichier, Édition, Affichage, Rechercher, Terminal, and Aide. The title bar of the window reads "esme@esme-VirtualBox: ~". The terminal prompt is "esme@esme-VirtualBox:~\$". The user then types "whatis clear" and presses Enter. The output shows the man page entry for the "clear" command: "clear (1) - clear the terminal screen". The window has standard Linux-style window controls (minimize, maximize, close) in the top right corner.

```
esme@esme-VirtualBox: ~
Fichier Édition Affichage Rechercher Terminal Aide
esme@esme-VirtualBox:~$ whatis clear
clear (1) - clear the terminal screen
esme@esme-VirtualBox:~$
```

# Command ls

## ls: list

- ❑ Usage: Lists files and directories in a directory.
- ❑ Simplified syntax:
  - ❑ ls [-a] [-l] [directory]
  - ❑ -a: Displays all files and directories in the directory, including those starting with '.' (hidden).
  - ❑ -l: Displays more information in a list format.
  - ❑ Example: ls -al /

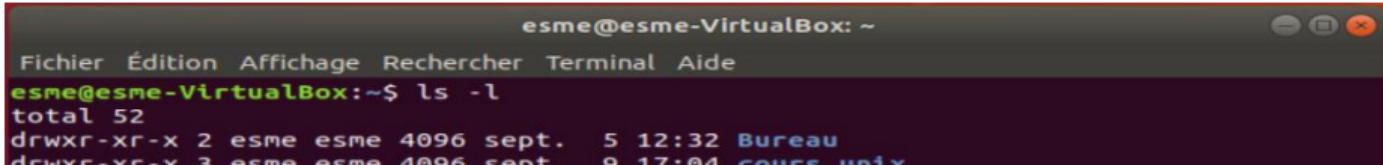


The screenshot shows a terminal window titled "esme@esme-VirtualBox: ~". The command "ls -al /" is run, displaying a detailed listing of files and directories in the root directory. The output includes file names, permissions, ownership (root/root), file size, modification date, and time, along with symbolic links like "initrd.img" and "initrd.img.old".

```
Fichier Édition Affichage Rechercher Terminal Aide
esme@esme-VirtualBox:~$ ls -al /
total 581684
drwxr-xr-x  24 root root      4096 sept.  4 09:40 .
drwxr-xr-x  24 root root      4096 sept.  4 09:40 ..
drwxr-xr-x   2 root root      4096 sept.  4 09:38 bin
drwxr-xr-x   3 root root      4096 sept.  4 09:41 boot
drwxrwxr-x   2 root root      4096 sept.  3 15:05 cdrom
drwxr-xr-x  18 root root     4160 sept. 30 10:09 dev
drwxr-xr-x 129 root root    12288 sept.  4 15:45 etc
drwxr-xr-x   3 root root      4096 sept.  3 15:07 home
lrwxrwxrwx   1 root root        32 sept.  4 09:40 initrd.img -> boot/initrd.img-
5.0.0-27-generic
lrwxrwxrwx   1 root root        32 sept.  3 15:01 initrd.img.old -> boot/initrd.
img-5.0.0-23-generic
```

## From Right to Left

- File or directory name
- Date and time of the last modification
- Size in bytes
- Group ownership
- Owner of the file/directory
- Number of links
- Access rights (nine characters)
- File type (one character)

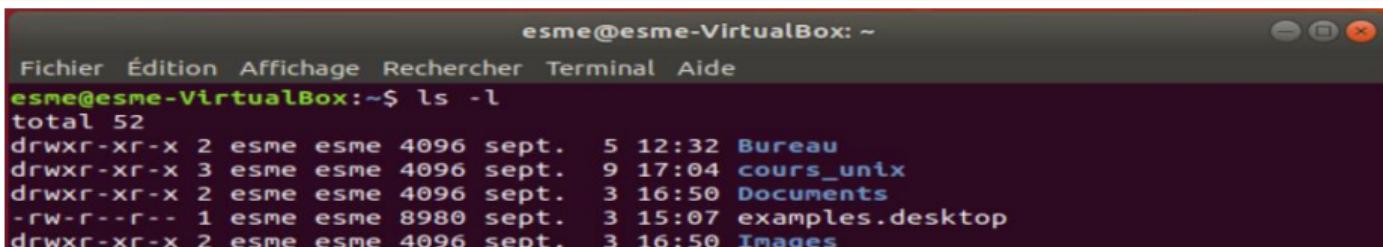


The screenshot shows a terminal window with a dark background and light-colored text. At the top, it says "esme@esme-VirtualBox: ~". Below that is a menu bar with "Fichier", "Édition", "Affichage", "Rechercher", "Terminal", and "Aide". The main area of the terminal displays the output of the "ls -l" command:

```
total 52
drwxr-xr-x 2 esme esme 4096 sept. 5 12:32 Bureau
drwxr-xr-x 3 esme esme 4096 sept. 8 17:04 COURS_unix
```

## File Types

- : Regular file
- d: Directory file
- l: Symbolic link (created with command ln)
- b: Block device (special file)
- c: Character device (special file)
- p: Named pipes or FIFO



The screenshot shows a terminal window with a dark background and light-colored text. At the top, the window title is "esme@esme-VirtualBox: ~". Below the title is a menu bar with French labels: Fichier, Édition, Affichage, Rechercher, Terminal, Aide. The main area of the terminal displays the output of the "ls -l" command:

```
esme@esme-VirtualBox:~$ ls -l
total 52
drwxr-xr-x 2 esme esme 4096 sept.  5 12:32 Bureau
drwxr-xr-x 3 esme esme 4096 sept.  9 17:04 cours_unix
drwxr-xr-x 2 esme esme 4096 sept.  3 16:50 Documents
-rw-r--r-- 1 esme esme 8980 sept.  3 15:07 examples.desktop
drwxr-xr-x 2 esme esme 4096 sept.  3 16:50 Images
```

## File Permissions

- ❑ Read from left to right, in groups of three
- ❑ For the user (user): u
- ❑ For group members (group): g
- ❑ For other users (others): o
- ❑ Permission types:
  - ❑ r: Read permission
  - ❑ w: Write permission
  - ❑ x: Execute or traverse directory permission
  - ❑ -: Permission denied

```
esme@esme-VirtualBox: ~
Fichier Édition Affichage Rechercher Terminal Aide
esme@esme-VirtualBox:~$ ls -l
total 52
drwxr-xr-x 2 esme esme 4096 sept. 5 12:32 Bureau
drwxr-xr-x 3 esme esme 4096 sept. 9 17:04 cours unix
```

## pwd: print working directory

- ❑ Usage: Displays the absolute name of the current directory.
- ❑ Simplified syntax: `pwd`



## cd: change directory

- ❑ Usage: Change the current directory with an absolute or relative path.
- ❑ Simplified syntax: cd [directory]
- ❑ Example: cd /tmp
- ❑ Note: cd without an argument sets the current directory to the user's home directory. A shortcut to the home directory of the current user is ~.



## mkdir: make directory

- ❑ Usage: Create a directory.
- ❑ Simplified syntax: `mkdir [-p] directory`
- ❑ `-p`: Create intermediate directories if necessary (creating entire branches).
- ❑ Example: `mkdir dir`
- ❑ Note: The directory name must not match an existing name.



## rmdir: remove directory

- ❑ Usage: Remove a directory if it's empty (except for '.' and '..').
- ❑ Simplified syntax: `rmdir [-p] directory`
- ❑ `-p`: Remove every directory within directory (destroying entire branches).
- ❑ Example: `rmdir dir`



## mv: move

- ❑ Usage: Rename or move files and/or directories.
- ❑ Simplified syntax: `mv source_name destination_name`
- ❑ It can be a file or a directory.
- ❑ Example: `mv file.txt /tmp`
- ❑ Note: The destination name should not already exist as a file or directory.



## cp: copy

- ❑ Usage: Copy files or directories.
- ❑ Simplified syntax: `cp [-r] source_name destination_name`
- ❑ It can be a file or a directory.
- ❑ `-r`: Copy a directory tree.
- ❑ Examples:
  - ❑ `cp file1.txt file2.txt`: Creates a copy of `file1.txt` named `file2.txt`.
  - ❑ `cp file1.txt file2.txt dir`: Copies `file1.txt` and `file2.txt` into the directory `dir`.
  - ❑ `cp -r tmp tmp1`: Copies the entire `tmp` directory into `tmp1`.

## rm: remove

- ❑ Usage: Delete files or directories.
- ❑ Simplified syntax: `rm [-r] file`
- ❑ It can be a file or a directory.
- ❑ `-r`: Delete a directory tree.
- ❑ Example: `rm -r dir`
- ❑ Be cautious when using this command; there is no trash bin in Linux.

