

# OPERATING SYSTEM: UNIX/LINUX



## Course 1

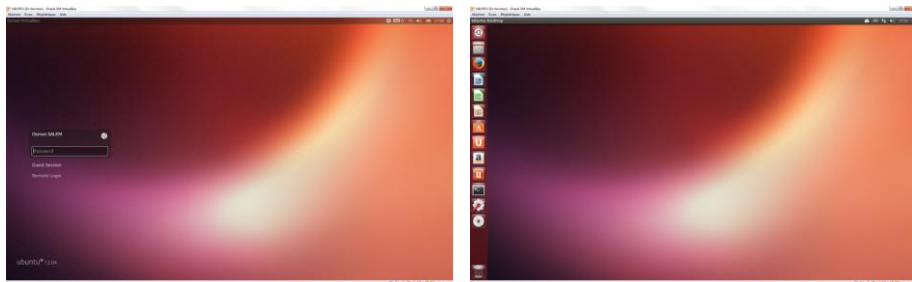
**Osman SALEM**  
**Maître de conférences - HDR**  
**[osman.salem@u-paris.fr](mailto:osman.salem@u-paris.fr)**



1



- Ubuntu
- You will use your laptop
  - Install Virtualbox/Vmware workstation
  - Install Ubuntu



2



## UNIX/LINUX OPERATING SYSTEM

---

- Introduction to Unix
- History of UNIX
- What is LINUX
- LINUX Distributions
- Unix OS Structure
- Unix File System
- Unix Directories, Files and Inodes
- Users, Groups and Permissions

3



## UNIX

---

- An operating system
- Developed at AT&T Bell Labs in the **1969**
- By **ken Thompson** and **Dennis Ritchie**
- Objective:
  - Portable OS written in C
  - Not in assembler
- Distributed as Open Source

4



- Command Line Interpreter
- GUIs (Window systems) are now available
- Unix becomes commercial & paying (outside AT&T)

5



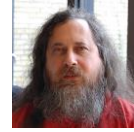
- Unix is a multi-user, multi-tasking operating system
- You can have many users logged into a system simultaneously, each running many programs
- It's the kernel's job to keep each process and user separate and to regulate access to system hardware, including cpu, memory, disk and other I/O devices

6

## History of UNIX

### ■ 1980

- Richard Stallman decides to create a clone of "Unix"
- With source code available (Open source)
- Creates Free software (most beautiful gift for the world!)
  - Open Source
  - You can look and modify the code
  - General Public License (GPL)
- Stallman want to create a derived version of "Unix"
  - GNU : GNU's Not Unix
  - Develop most of Unix Commands (GCC: C compiler)
- But they were late to develop the Kernel
  - Communications between software and hardware
  - Memory management, Process management, CPU usage, hard disk, network card, etc.



Richard Stallman

7

## History of LINUX

### ■ Linus Torvals in 1991

- Has created the kernel using Minix
- Publish in web forum
- Like Free Software (General Public License)
- Error in naming
  - Kernel : Linux à la place de Linus
- Invite other to join his project



Linus Torvalds

### ■ Andrew Tanenbaum

- Computer Science professor emeritus in Amsterdam
- Author of **Minix** => Teaching the OS
- **Minix**: Unix-like operating system for teaching purposes



Andrew Tanenbaum

### ■ Like kernel is the most important part

- Linux + GNU = *Linux*
- *Stallman* ask to call it: *GNU /Linux*
  - It's so late, because it was widely known as Linux

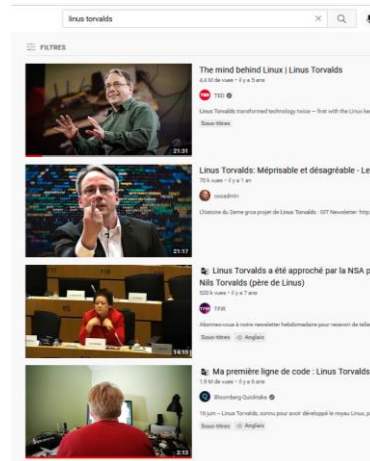


Richard Stallman

8

## Version of kernel

- Kernel version with the command:
  - `uname -r`
    - It's so late, because it was widely known as Linux
    - Kernel version: "x.y.z"
      - if "y" is even => stable version
      - If "y" is odd=> development version (under progress and unstable)
      - `$uname -r => 3.8.0-19-generic`
      - `$cat /etc/lsb-release`: ubuntu version, independent of kernel version
      - `$lsb_release -a` et `cat /etc/issue`



Linus Torvalds

9

## Unix vs. LINUX

- Unix was the predecessor of Linux
- Linux is a variant of Unix
  - So is Mac OS X, so much of this tutorial applies to Macs as well
  - Linux is open source
  - Linux is **free**
  - It's fully **customizable**
  - It's **stable** (i.e. it almost never crashes)
  - These characteristics make it an ideal OS for programmers and scientists

10

## LINUX Distributions

- Mandrake: <http://www.mandrakesoft.com/>
- RedHat: <http://www.redhat.com/>
- Fedora: <http://www.fedora-fr.org/>
- CentOS : <http://www.centos.org/>
- Debian: <http://www.debian.org/> (very secure)
- Ubuntu: <http://www.ubuntu-fr.org/>
- SuSE/Novell: <http://www.suse.com/>
- Kali: <https://www.kali.org/>
- etc. (or distrowatch.com)



11

## *Installation*

12

## Installation

---

- Root password
- Network Devices
- Services
- Users
- Hardware

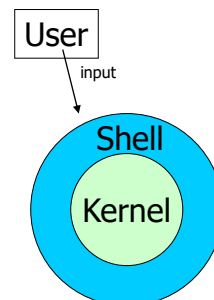
13

## UNIX Structure

---

### Shell

- Command line interpreter
- Shell is an interface between user and kernel
- Shell interprets your input as commands and pass them to kernel



14

## Connecting to a Unix/Linux system

### ■ Attention:

- Linux is case **sensitive**
- **file.txt** is different from ***FILE.txt***

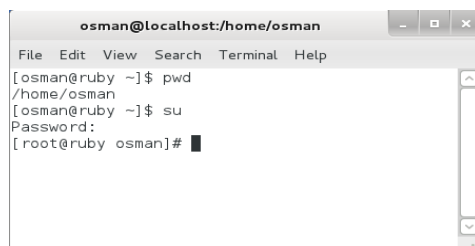
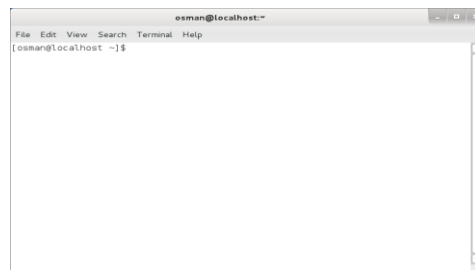
*Really ! Why they  
are different ?*



15

## Terminal

- How to open a terminal ?



**hostname**

**location in file system**

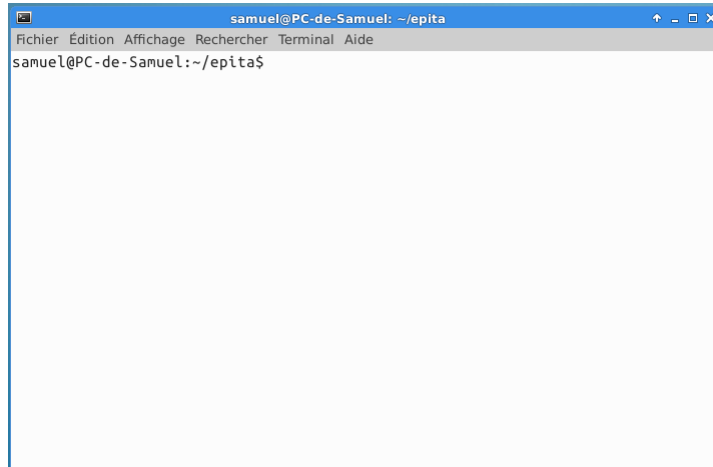
**user indicator, \$ = regular user**

16



## Connecting to a Unix/Linux system

- Open a terminal:

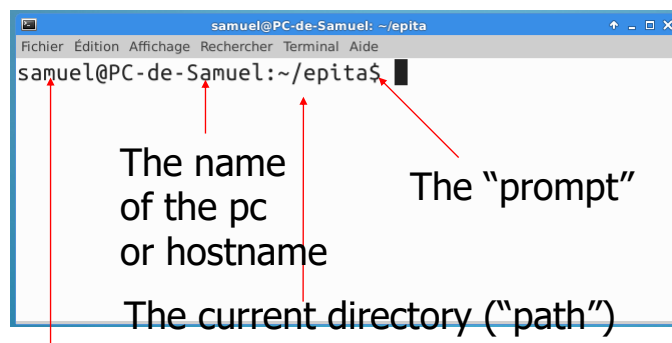


```
samuel@PC-de-Samuel: ~/epita
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
samuel@PC-de-Samuel:~/epita$
```

17

## Connecting to a Unix/Linux system

- Open a terminal:



```
samuel@PC-de-Samuel: ~/epita
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
samuel@PC-de-Samuel:~/epita$
```

The username

The name of the pc or hostname

The current directory ("path")

The "prompt"

18



## Shell commands

### ■ ***Some commands***

- date
- who am i : account used
- who: list of connected users to the system
- cal: calender
- uname: OS & kernel version
- id: identity of user
- su: switch user
- script -a => exit: save in file name typescript
- file *filename*: the type of the file

19



## Shell commands

### ■ ***Some commands***

- date
- ls, cp, rm, mv, ln, pwd, cd, mkdir, find, cat, more, less, grep, sort, tail, head, wc, whereis, alias, unalias, type
- chgrp, df, file, tac, rev, tr, date, clear, diff, passwd, who, whoami
- Wildcards symbol: \*, ?, []

20



## UNIX Shell

- The shell sits between you and the operating system, acting as a command interpreter.
- It reads your terminal input and translates the commands into actions taken by the system.
- When you log into the system you are given a default shell.
- When the shell starts up it reads its startup files and may set environment variables, command search paths, and command aliases, and executes any commands specified in these files.
  1. /etc/profile  
Then attempts to read and execute (but stop after the first file is found)
  1. .bash\_profile (fedora, redhat, centos, etc.)
  2. .bash\_login
  3. .profile (debian, ubuntu)

21



## UNIX Shell

- Shell sessions can be divided into 2: login shells and non-login shells
  - Login shell : ctrl-alt-F1 =>F6 & session on remote hosts (ssh, telnet)
  - Non-login shell : terminal session on the GUI desktop
- Startup files divided into 2 groups
- Global files in /etc and applied to all users
  - Non login: /etc/bash.bashrc
- Local files in user's home directory
  - Non login: .bashrc
- .bash\_profile or .profile : **source** .bashrc file: read this file during startup of login and non-login shells, so putting your changes in .bashrc is usually a safe bet

22



## Customize your shell

- **# nano /home/login/.bashrc**
  - alias ll="ls | less"
  - alias x= "ls -l"
  - alias n= "nano"
- Restart your terminal
- Or
- *source .bashrc*

23



## Help!

- Whenever you need help with a command type "man" and the command name
- The **man** command is used to display the manual entry associated with **word** entered as argument.
- The **-k** option is used to display a list of manual entries that contain entered **keyword**.

**man** [chapter] word

**man -k** keyword

24



## Help!

```
wiche@zhome:~/linux_tutorial$ man
What manual page do you want?
zhome:~/linux_tutorial$ man echo
zhome:~/linux_tutorial$
```

25



## Help!

```
wiche@zhome:~$ man echo
ECHO(1)                                User Commands                                ECHO(1)

NAME
    echo - display a line of text

SYNOPSIS
    echo [OPTION]... [STRING]...

DESCRIPTION
    NOTE: your shell may have its own version of echo
    which will supercede the version described here.
    Please refer to your shell's documentation for
    details about the options it supports.

    Echo the STRING(s) to standard output.

    -n      do not output the trailing newline
lines 1-19
```

26

# Help!

```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ man
What manual page do you want?
zhome:~/linux_tutorial$ man echo
zhome:~/linux_tutorial$ echo hello world
hello world
zhome:~/linux_tutorial$
```

27

# UNIX File System

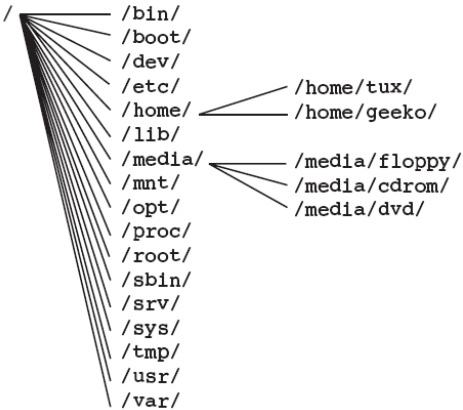


Figure 5-1

28




## File System

---

- The Unix file system looks like an inverted tree structure.
- You start with the root directory, denoted by /, at the top and work down through sub-directories underneath it.
- Root Directory /

29



## Structure of Standard Directories in Unix/Linux

---

- / The ancestor of all directories on the system; all other directories are subdirectories of this directory, either directly or through other subdirectories.
- /bin Essential tools and other programs (or binaries).
- /dev Files representing the system's various hardware devices. For example, you use the file `/dev/cdrom` to access the CD-ROM drive.
- /etc Miscellaneous system configuration files, startup files, etc.

30



## Structure of Standard Directories in Unix/Linux

---

- /home The home directories for all of the system's users.
- /lib Essential system library files used by tools in `/bin`.
- /proc Files that give information about current system processes.
- /root The superuser's home directory, whose username is root. (In the past, the home directory for the superuser was simply `/`; later, `/root` was adopted for this purpose to reduce clutter in `/`.)

31



## Structure of Standard Directories in Unix/Linux

---

- /sbin Essential system administrator tools, or system binaries.
- /tmp Temporary files.
- /usr Subdirectories with files related to user tools and applications.

32





# Essential Binaries for Use by All Users (/bin/)

Table 5-1

| File       | Description          |
|------------|----------------------|
| /bin/bash  | The bash shell       |
| /bin/cat   | Display files        |
| /bin/cp    | Copy files           |
| /bin/dd    | Copy files byte-wise |
| /bin/gzip  | Compress files       |
| /bin/mount | Mount file systems   |
| /bin/rm    | Delete files         |
| /bin/vi    | vi editor            |



# Libraries (/lib/)

- **Shared libraries** are removed from the actual program, stored in the system, and only called up when the program runs
- The directory /lib/ contains the libraries that are used by programs in the directories /bin/ and /sbin/
- The kernel modules (hardware drivers not compiled into the kernel) are located in the directory /lib/modules/
- You can find additional libraries below the directory /usr/



## Mount Points for Removable Media (/media/\*)

---

- Linux creates directories such as the following in the directory /media/ (depending on your hardware) for mounting removable media:
  - **/media/cdrom/**
  - **/media/cdrecorder/**
  - **/media/dvd/**
  - **/media/floppy/**

35



## System Binaries (/sbin/)

---

- Contains important system administration programs
- Programs in /sbin/ can also, as a rule, be run by normal users, but only to display configured values

36



## Temporary Area (/tmp/)

- Various programs create temporary files that are stored in /tmp/ until they are deleted

37



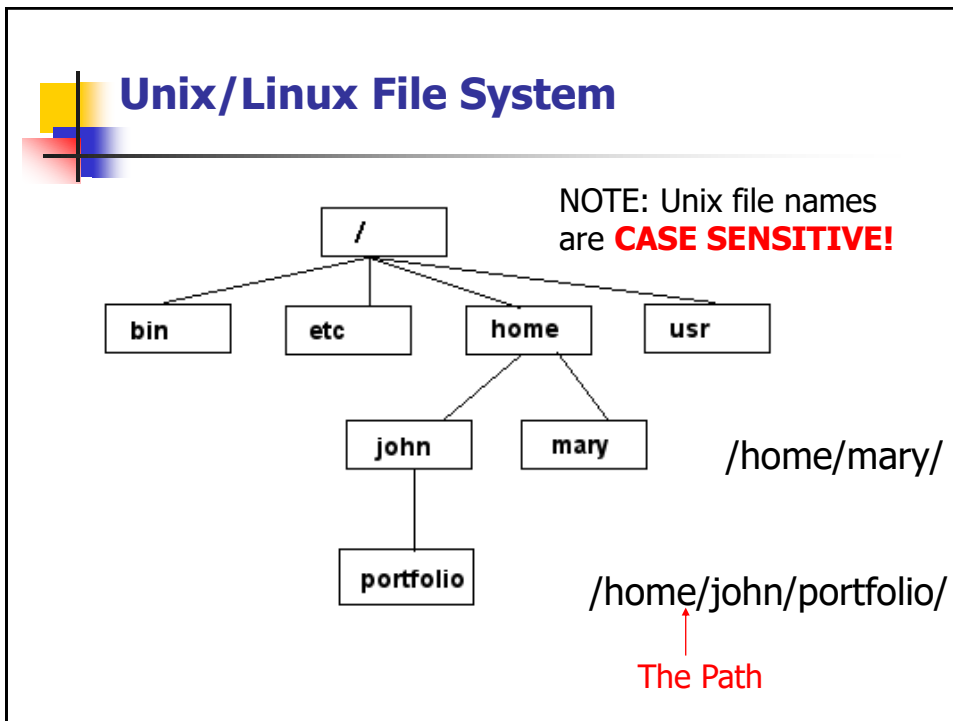
## Variable Files (/var/)

- This directory and its subdirectories contain files that can be modified while the system is running
  - File in printing

Table 5-7

| Directory   | Description  |
|-------------|--|
| /var/lib/   | Variable libraries (such as databases for the commands locate and rpm) |
| /var/log/   | Log files for most services  |
| /var/run/   | Files with information on running processes                            |
| /var/spool/ | Directory for queues (printers, e-mail)                                |
| /var/lock/  | Lock files to protect devices from multiple use                        |

38



39

## Relative Path

- Relative to your current location
  - . : your current location
  - .. : one directory above your current location
  - pwd: gives you your current location
- Example
  - ls ./linux : lists the content of the dir linux
  - ls ../../ : lists everything that is two dir higher

40

## User Directories (/home/)

- Every user on a Linux system has his own area in which to create and remove files: its home directory
- Individual configuration files can be found in the user's home directory

Table 5-4

| File          | Description                             |
|---------------|---|
| .profile      | User's private login script             |
| .bashrc       | Configuration file for bash             |
| .bash_history | List of commands previously run in bash |

- If there are no special settings, the home directories of all users are located beneath /home/
- The home directory of a user be addressed via "~"

41

## UNIX File System

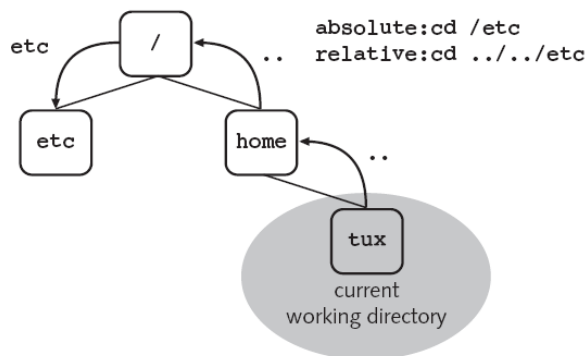


Figure 5-2

42

## Command: pwd

- To find your current path use "pwd"

```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ pwd
/fs/zhome05/wiehe/linux_tutorial
zhome:~/linux_tutorial$
```

43

## Command: cd

- To change to a specific directory use "cd"

```
wiehe@zhome:~/linux_tutorial
zhome:~$ pwd
/fs/zhome05/wiehe
zhome:~$ cd /fs/zhome05/wiehe/linux_tutorial/
zhome:~/linux_tutorial$ pwd
/fs/zhome05/wiehe/linux_tutorial
zhome:~/linux_tutorial$
```

44



## Command: cd

- “~” is the location of your home directory

```
wiehe@zhome:~  
zhome:~/linux_tutorial$ pwd  
/fs/zhome05/wiehe/linux_tutorial  
zhome:~/linux_tutorial$ cd ~  
zhome:~$ pwd  
/fs/zhome05/wiehe  
zhome:~$
```

45



## Command: cd

- “..” is the location of the directory below current one

```
wiehe@zhome:~  
zhome:~/linux_tutorial$ pwd  
/fs/zhome05/wiehe/linux_tutorial  
zhome:~/linux_tutorial$ cd ..  
zhome:~$ pwd  
/fs/zhome05/wiehe  
zhome:~$
```

46



## Moving in Directories

- `cd try_it`
  - Changes the directory to `try_it`
- `pwd`
  - Prints present Working Directory (e.g. `/home/smith/try_it`)
- `cd ..`
  - Move to superior directory
- `pwd`
  - Prints `/home/smith`
- `cd /home`
  - The absolute path
- `pwd`
  - Prints `/home`

47



## Moving in Directories

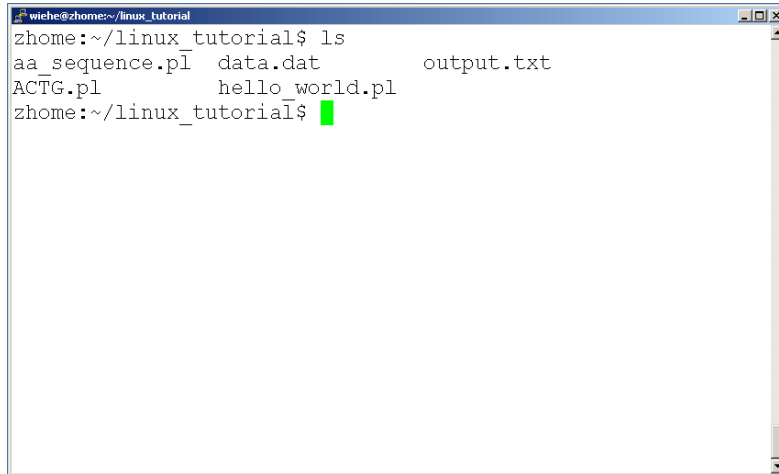
- **`cd`**
  - *The system is returned to the user home directory*
  - `cd ../../rep1/srep1; cd ~rep1/srep1; cd; cd ~toto/public-html`
- `cd $HOME <=> cd <=> cd ~`
- `echo $USER`
- `whoami`
- `w & who`

48



## Command: ls

- To list the files in the current directory use "ls"

A terminal window titled 'wiehe@zhome:~/linux\_tutorial' showing the command 'ls' and its output. The output lists files in three columns: 'aa\_sequence.pl', 'data.dat', and 'output.txt' on the first line, and 'ACTG.pl' and 'hello\_world.pl' on the second line.

```
wiehe@zhome:~/linux_tutorial$ ls
aa_sequence.pl  data.dat      output.txt
ACTG.pl        hello_world.pl
zhome:~/linux_tutorial$
```

49

## Listing the Content of a Directory

- `ls` is used to list the contents of a directory.
- If the command `ls` is written with parameter `-l`
  - then the command lists contents of the working directory with details. Example:
  - `$ ls -l`

50



## Command: ls

- `ls` has many options
  - `-l` long list (displays lots of info)
  - `-t` sort by modification time
  - `-S` sort by size
  - `-h` list file sizes in human readable format
  - `-r` reverse the order
  - `-a` list all files (with hidden files)
  - `-A` same as `-a` minus directories `."` and `.."`
  - `-d` list only directories
  - `-g` do not list owner (similar to `ls -l` but without owner)
- "***man ls***" for more options
- Options can be combined: "`ls -ltr`"