

Skills acquired in this Lab

- Remote connection and administration of a computer using SSH.
- The concept of command prompt: the bash shell.
- Knowledge and practice of the basic commands of the Linux environment.

Why do we need to learn Linux commands ?

As explained in the beginning of this course, the objective is to develop a connected device that will do some cool stuff like controlling the lights, printing the humidity and temperature, etc. In order to do so, we will use the Raspberry Pi. The Raspberry runs on a Linux operating system (like your PC on windows or Mac). For that, and in order to work in a Linux OS (operating system), we will practice some basic Linux commands!

Terms and conditions

- Duration 2 Hours, alone.
- Personal computer required. For IT security reasons, this Lab can only be performed if your computer is connected to the Wi-Fi network called "Yncrea_LAB".
Password: 813nV3nue@

Introduction

In this Lab, you will connect to a remote device located in HEI's premises: a server running on Linux.

A server is a computer capable of supporting several simultaneous users.

Linux is a system competing with Windows that works on very powerful servers and on many of the connected devices. Even if Linux offers a graphical user interface, you will use the console mode.

Console mode looks visually similar to Microsoft's MS-DOS mode (white text lines on a black background). However, it is a complete Linux in multi-tasking and multi-user mode.

This mode is particularly useful in 3 specific cases:

- We want to have as much power as possible (the graphic system consumes in processor resources and memory)
- The graphic system may not be available for various reasons (system without display, failure, terminal in text mode...)
- Remote management of a system.

Contrary to what you might think, the console mode is “simpler” than the graphic mode. Almost anything that can be done in graphic mode can be done in console mode. With the exception of graphic applications obviously (drawing, photos, videos, etc.).

The command prompt

The command prompt is the input by which a shell lets you enter commands that allow you to access the internal features to the operating system. Unix operating systems most often have a shell.

Originally, the default shell was sh, which gave rise to many variants, including csh, extended in tcsh, or ksh, or rc... However, today bash, inspired by sh, ksh, and csh, is the most common shell, although there are other shells, such as zsh, or ash.

The bash prompt looks something like this:

username@machinename:currentdirectory\$

The bash prompt allows you to know at a glance which user account you are using, to which machine you are connected and in which directory you are located.

SSH and the command line

Until now, you have always used your computer the same way: you were in front of it. You were physically next to your machine, for example, you pressed the “button Power” to turn it on.

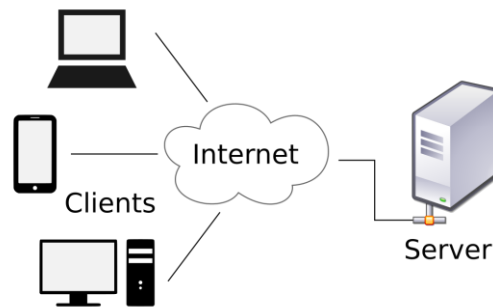
So far, nothing new. However, one of the great strengths of Linux is that it can be used even if you are hundreds of kilometers away from the machine. This operation dates back to the time when it was necessary to administer machines remotely.

Today, if I live in Paris, I can “easily” control a Linux computer located in Tokyo, Japan, along with another computer located in the far reaches of Nevada, United States. I can even order the Tokyo computer to send a file to the Nevada computer. This kind of manipulation is now possible via Internet, and it is done every day!

The people who manage Linux machines, often remotely, are called **system administrators**. Fortunately, it is not necessary to be physically present next to the machine to work on it!

You imagine, having to buy a round-trip ticket to Tokyo just because you need to install a program on a server....

The PC that connects to the server is called the client.



To communicate with each other in a network, two computers must use the same protocol.

It is like a language: to speak with someone, you have to speak the same language like him, or you will not understand each other.

There are many protocols, we will use SSH.

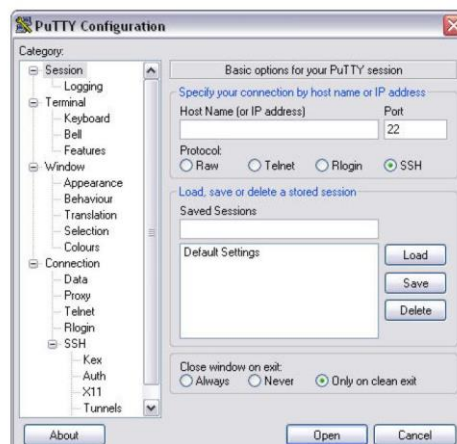
SSH for Secure SHell encrypts the exchanges between the client and the server.

- If you have an Apple Mac or a PC running Linux, you do not need to install any particular software for SSH, it is already installed by default.
All you have to do is to launch the Terminal application.
Then type: `ssh userXX@10.34.161.21` with XX given by your teacher.
The password is `user-hei`. You will not see your typing on the screen, it is normal.
- For Windows PCs you will need to download PuTTY.
Go to <http://www.putty.org> and click on the "You can download PuTTY here."
Then download the suitable file for your OS (32 or 64 bits)

Not sure whether you want the 32-bit or the 64-bit version?

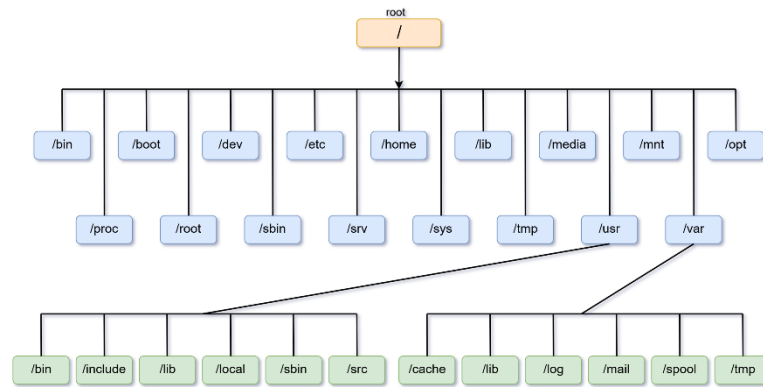
Go to My Computer (Ce PC) -> right click -> Properties.

- Start the installation. At the end, start PuTTY.
In the following window:



- Specify as hostname: **10.34.161.21** and click on Open.
- Accept the server's fingerprint.
- Then enter the userXX user name with XX given by your teacher when the machine asks you for the login.

The password is **user-hei**. You will not see your typing on the screen, it is normal.



Linux File System

Basic commands of the Linux environment

- **man & --help** — To know more about a command and how to use it, use the **man** command. It shows the manual pages of the command. For example, “**man cd**” shows the manual pages of the **cd** command. Typing in the command name and the argument helps it show which ways the command can be used (e.g., **cd --help**).
- Type **man ls** in the command line to get help on the **ls** command.

Now for the rest of this Lab, you should type on google: **man <Command>** , where the **<command>** is given below

1. Command: **ls** Displays the content of a directory

- a. Type: **ls**
What do you see on the screen?

2. Command: **mkdir** Create a new directory

- a. Type: **mkdir toto**
- b. Type: **ls**
What do you see on the screen?

What is then the utility of the command **mkdir**?

3. **Command: pwd** Displays the name of the current directory

- a. Type: **pwd**

In which directory are you located?

What is then the utility of the command **pwd**?

➤ You will notice that under Linux, the root of the disk is symbolized by the character **/**. It is also the separator between directories. (Under Windows, the root of the disk is a letter followed by two points, and the separator is the character ****. Example: C:\Windows\tmp)

4. **Command: cd** Changes the working directory

- a. Type: **cd toto**

What do you realize?

- b. Type: **pwd**

What do you see on the screen?

What is then the utility of **cd**?

- c. Type: **cd /**

Where are you located now in the tree structure?

- d. Type: **cd /home/userXX/toto** where XX is your user number

- e. Type: **cd ~**

What do you think is then the difference between **cd ~** and **cd /** ?

- f. Type: **cd ..**

What do you realize?

Absolute and Relative path:

In order to allow access to a file and/or directory with **cd** or anywhere else in Linux, we can use:

- The absolute path (slash as a prefix to the path)
- The relative path (no initial slash)

An **absolute path** is defined as the specifying the location of a file or directory from the root directory (**/**). In other words, we can say absolute path is a complete path from start of actual file system from **/** directory.

- Type: **cd /home/userXX/toto** where XX is your user number
In this example, we use /home/userXX/toto to change directory to toto.
/home/userXX/toto is an absolute path.

A **relative path** is defined as the path related to the present working directory (pwd). Suppose I am located in /home/userXX and I want to change directory to /home/userXX/toto. I can use relative path concept to change directory to toto.

- Type: **cd toto**
In this example, we use a relative path to change directory to toto.
In this writing, the current directory is called . (the dot character).
The parent directory is called .. (the dot character repeated twice).
- a. Go back to your home directory. Reminder : **cd ~**
- b. Type: **mkdir toto2**
What is the absolute path in order to get to toto2?

What is the relative path to go to toto2 from the current directory?

- c. Go to the toto2 directory.
Give four different commands to return to ~

Test them. Are they absolute paths? Relative paths?

5. Command: cat

- a. Go to the /etc directory
- b. Type: **ls -d s***
What does this command do?

What is the common point between all of the listed files?

- c. Type: **cat services**
Then press (ctrl-c) to stop
- d. Type: **cat services | more**
Press ENTER
What do you see?

What is the utility of the command "more" ?

- e. Type: **cat -n passwd**
What do you see? What is the utility of the -n option?

6. **Command: cp** Copy a source file to another file

- a. Go to the directory /etc/X11 (do pwd to check)
Note some of the existing files

- b. Type: **cp xorg.conf ~**
- c. Go to your home directory.
- d. Type: **ls**
What do you notice?

What is then the utility of the command cp?

- e. Type: **man cp**
What is the command to copy a directory with its content? Test it.

7. **Command: mv**

- a. Go to your home directory
- b. Type : **touch coucou.test**
- c. Type : **ls**
What do you realize?

What is the utility of the command touch?

- d. Type: **mv coucou.test coucou.test.new**
- e. Type: **ls**
What do you realize?

What is the result of this command?

- f. Move the file coucou.test.new to the toto directory.
Write down the command that you used.

8. **Command: rm** Delete a target file or directory.

- a. Test the **rm** command, delete the file coucou.test.new
Write down the command that you used.
- b. Type: **man rm**
What is the command to remove a directory with its content? Test it.