## Report of Project: Linux\_toolbox Operating System and Linux Administration

Table des matières

[Report of Project: Linux\_toolbox Operating System and Linux Administration 1](#_Toc54097423)

[Introduction: 1](#_Toc54097424)

[Summary of the project: 1](#_Toc54097425)

[Conception of the archive structure: 1](#_Toc54097426)

[scrip: 2](#_Toc54097427)

[Client oriented script: 3](#_Toc54097428)

[Server script: 3](#_Toc54097429)

[Conclusion: 3](#_Toc54097430)

### Introduction:

This project aims to make us apply everything we learn in the Operating System and Linux Administration course and make it something. We will be doing this project using the same pair of students as the TP.

### Summary of the project:

The project asks us to deliver a toolbox composed of set of command, file systems and configuration file. Those will serve as a base to create and manage a file system for students and teachers for their courses.   
In it we will see the storage of every bit of information required for a student and a teacher for a course. The handling of editing it. And the access of that information remotely.

### Conception of the archive structure:

First, we started to read every question before writing any script.   
By doing so we where able to add every entry we were asked to handle.

We also where able to describe the directory tree:[1]

By extracting the information, we need to handle we made the fooling structure for the semester.info [2]

The semester.info just give the directory structure and not every file in it. So we just decided to store information based on modules. By doing so we know the common structure of every module above them. That also means that a UE cannot exist without a module. Then we fill the module with the number, of course, that will be included in the module. Last, we store the year and the semester in with they are attached. By doing so we reduce a maximum, the number of entries and redundancy using this kind of structure register and we avoid using complex way of storing a tree.

Adding to this structure, we also store important data needed by some user and asked to be stored live the coefficient of the UE, the teacher in charge of this module with their mail. And the tie of evaluation in this module with also it coefficient.

We also made the structure for the semester.conf which give the information about who can access a file and what is their right with it. [3]

### scrip:

Question 1:   
The following schema [4] give us an understanding of what script is executed after lunching the first script and the order in with they will be executed.

Script summary:

-User selection

When lunched the user is asked what his function is. He can choose between three choices (Administrator, Teacher and Student).  
After that and I’d been asked to be sure that he is really a user of that type.\*\*\*(verification process to be made)\*\*\*  
Then it is asked to the user how he wants to create his semester directory. Two options are given to him (Import a .info or manually enter each directory we want to create.

After that, every choice is passed in argument of another script.

-Import semester sheet

Given the path, the script we will look for it. Check if it is in the right format and then verify if the directory does not already exist.

-Create semester sheet

The script will ask every question to create an entry for a .info.

Verification on the semester will be made to be sure not to create 2 times the same.   
After that he will create the directory tree if no error found.

#### Question 2:

The following schema [5] give us an understanding of what script is executed after lunching the first script and the order in with they will be executed.

Script summary:

-User selection

When lunched the user is asked what his function is. He can choose between three choices (Administrator, Teacher and Student).  
After that and I’d been asked to be sure that he is really a user of that type.\*\*\*(verification process to be made)\*\*\*  
Then it is asked to the user how he wants to create his semester directory. Two options are given to him, add or delete a directory.

After that, every choice is passed in argument of another script.

-Add directory

Given a path, if the parent exists a new directory can be created thanks to the name given in argument. Before creating it, we will check in the .conf if the user has the right to do this action.

Then it is given the option to the user to go to “Add directory” or “Delete directory” passing the argument already existing. If not the script terminate

-Delete directory

Given a path, if the directory exists and the .conf give the user the right to do this action, the directory will be deleted.

Then it is given the option to the user to go to “Add directory” or “Delete directory” passing the argument already existing. If not, the script terminates

#### Question 3:

The structure of the question one is taken as a based.

### Client oriented script:

### Server script:

Conclusion:   
  
Annex:

[1]

Semester

UE

module

TP\_1

TD\_1

CM\_1

Miscellaneous

School\_Directory

Year

Year

Semester

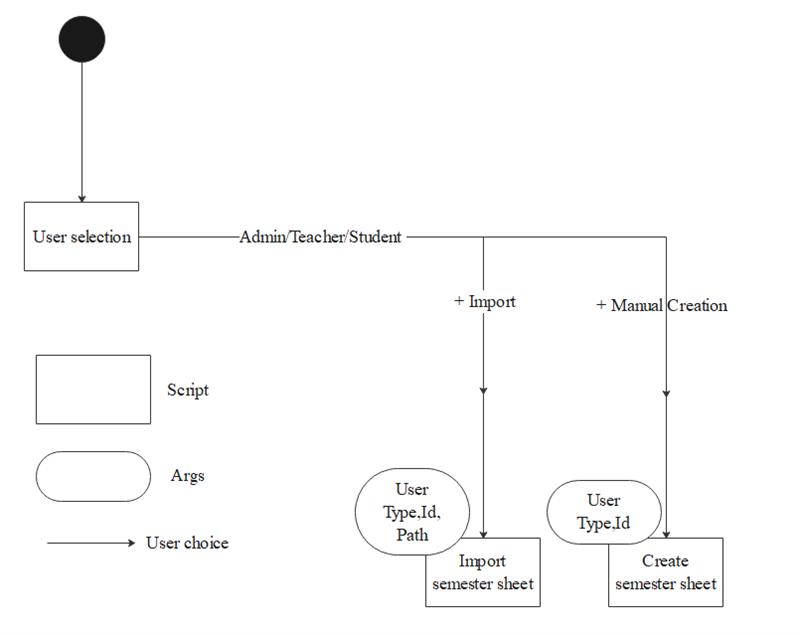
[2]



[3]



[4]



[5]

