

# Project

## Operating System and Linux Administration

This project aims to apply the knowledge and skills acquired in the previous sessions and to learn new skills related to the design, proposition, requirements, specifications and teamwork.

In this project, your teacher will play both roles, those of the client and the supervisor. So, do not hesitate to ask questions, to discuss your design and proposition, and to ask for help in case of difficulty. Note that a large range of freedom is given, which requires you to be proactive. This corresponds to real situations in which an engineer could be.

To approach the subject well and progress in the project you must first read and fully understand the subject, then start by proposing a design and discussing it with your teacher, find the different commands and tools necessary for your work, define the roles and tasks of each team member and write test scripts to validate your program at each step.

Each group must submit a report describing the design and the work done as well as all the different scripts. The submission deadline will be defined according to the teams' progression.

In this project, we want to propose a toolbox (set of commands, file system and configuration files) that allows students and teachers to manage their courses: content, progressions, grades, questions, sharing, collaboration etc. This need you to work with your team to meet the needs bellow. You have to propose a script/command for each need. Do not forget to manage errors and exceptional cases, and to provide messages and help panels. You have to distinguish the role of the administrator that manage your program and the user for whom all the underlying operations are transparent.

1. Your program must first provide the possibility for the students to create a semester using two methods:
  - Import: by taking as input a file describing the semester (semester sheet)
  - Manually: by asking the user to enter the information related to the semester content

For this first feature you must propose: for the first method, a *semester sheet* model containing the names of the teaching units (UE), modules, associated teachers and their emails, coefficient of each module, the teaching methods (TP, TD, CM...) as well as the different evaluation methods (of the same module) and their coefficients. For the second method, an interaction procedure with the user allowing to enter the information (the same as the first method).

This process must create:

- a. A file tree structure containing directories corresponding to the different teaching units, modules and teaching types.
  - b. An information file `semester.info` that could be read by all users and editable only by root. This file contains information related to the semester (to be defined as the progress is made), at this step, it corresponds to the semester sheet.
2. The user should be able to add/delete (update) courses, UE, TP etc. this should update `semester.info`  
Note that we are talking about organization directories; content files are added and removed by the user using the standard Unix commands.
3. By default, at the creation of a semester, your program must associate the user to its semester; this information (of association) must be maintained in the file `semester.conf` (you create). The user should be able to associate other users and groups to a given part (specific cours, TP, TD...). A user needs to have all permissions for all his semester content. The other associated users must have all permissions for their parts (folders and files) and no permission at all for the other semester content.
4. A user must be able to add a work “devoir” related to a given course (TD/TP/Project...) and to follow its progression. At the creation, the program creates a corresponding directory and associates a progression percentage value. You must use information and configuration files to maintain this information. This value must be editable by the users as the work progresses.
5. Also, users should be able to follow the courses evaluation grades (notes). For that, your program must associate a grade value (/20) to each assessable part (TD, TP, exam...). So far, the grades are entered and edited manually. You can modify your information and configuration files to be consistent with the features.
6. Use `gnuplot` tool to generate graphical visualization of work progress and grades. This should be opened in an Internet browser (automatically generated html) or PDF. The `gnuplot` generated files should be invisible to the user and deleted after.
7. The user should be able to write down questions and remarks related to a course. For that, the program needs to create a question/remark text file in each part and feed it at each user request.
8. A collaborator (associated user/group) can send/receive files remotely to/from the user machine. You need for this question to propose a method to find the right directory according to the collaborator request. You can add new information/configuration files.

9. Add a server containing :
  - a. Different semester sheets
  - b. A directory for each teacher (who correspond to defined users) containing a directory for each course, TP, TD ... (file tree) with a
  - c. Files containing students grades

It is up to you to define the files structures, permissions, directories ...

10. Modify the semester creation and the update command to take into account the server method. The address and the information related to the server must be defined in the configuration file and the operations must be transparent to the user. The user asks only for the list of the existing semesters then create or update one. Your program then manage the transmission of the files. In the case of update, your program should keep the existing parts and add only the new ones.
11. A user should be able to update all the grades (from the server) or set a frequency for an automatic periodic update. An email is sent (use `mail` or `sendmail`) to the student if a new grade is added.
12. The administrator can set a frequency for an automatic upload of questions and remarks (to each specific teacher directory). If a new question is detected an email is sent to the teacher indicating the new questions.
13. A student can also submit his work (devoir) result. Your program upload then an archive of the work to the teacher directory on the server and send an email.
14. A semester can be saved with its content (use archive files) and transferred on another client machine.
15. Propose a new feature.