Systems Architecture

Laboratory
Directory management and information retrival

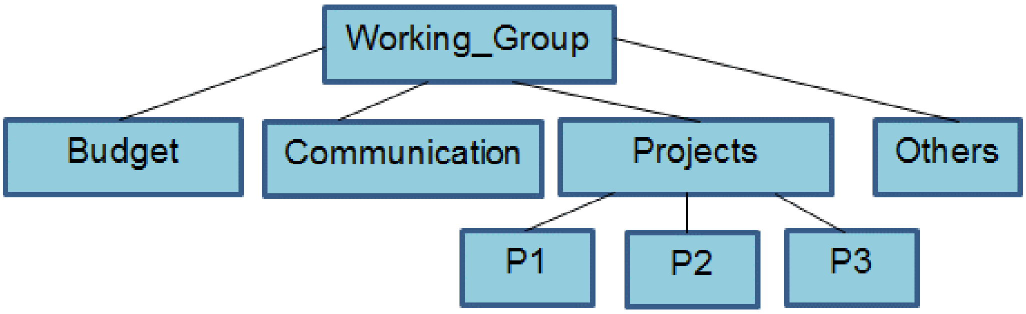
Objective: Learn how to manage information in Linux environment.

Exercise 1

Creation of a set of folders and files.

In this task, you must execute a set of Linux commands for the creation of some directories and files with specific permissions.

1. You must create the folder or directory structure that is indicated in the figure, and the directory“Working_Group” must be under the path /home/teleco. Execute the command sequence that is needed for that, taking into account that the directory initial location where you are in the system can be whatever, and the command sequence must work correctly with independency of the initial directory where you are located. In order to do this, you must use the commands **mkdir** and **cd**. Another mandatory requirement is that you must use the command **mkdir** using an absolute path in one of the commands but also a relative path in another command.



2. Next, you must assign the proper permissions to each directory, so that you can have control about who can access these directories and with what permissions. All the directories must have read and execute permissions for the members of the owner's Linux group, and in addition write permissions for the directory owner, and none permissions for the rest. Moreover, there is an exception, for the directory Budget, permissions must be read, write and execution for the directory owner, and none permissions for the rest, because only the project boss is allowed to operate with the budget files. Execute the command sequence that is needed for assigning all of these permissions, taking into account that the directory initial location where you are in the

system can be whatever, and the command sequence must work correctly with independency of the initial directory where you are located. In order to do this, you must use the commands **chmod** and **cd**. Another mandatory requirement is that you must use the command **chmod** using an absolute path in one of the commands but also a relative path in another command.

3. Next, you must create a file description.txt under the directory P1 which must include inside the file the text "Project Description". In order to achieve it, it is mandatory to use the command echo and the output redirection.
4. Copy the file description.txt from the folder P1 to the folders P2 and P3, so that the file will also be present in these other folders. In order to achieve it, you must use the command **cp**.
5. Copy the file description.txt also to the directory Budget, and to the directory Communication using the command **cp**. Next, change the name of the file that is under the directory Budget so that it is renamed into budget.ax, using the command mv. Next, replace completely the file text by the following text: "Budget file". In order to achieve it, another requirement is that you must use the command **echo** and the output redirection.
6. Move the file description.txt that is under the directory Communication, to the directory Others, and at the same time change its name so that the new name must be others.txt. For all of this, you must use the command mv. Next, replace completely the file text by the following text: "Other issues". In order to achieve it, another requirement is that you must use the command echo and the output redirection.
7. Finally, put all the sequence of commands that you have done in a file called creation.sh with a text editor and save it. You must write a line for each one of the commands and in the same order that you did in this exercise. Furthermore, add the following text in the first line of the file:

```
#!/bin/bash
```

Assign permissions for the execution of the new created file. This new file that contains a command sequence is called script. If you execute this file from a command interpreter with ./creation.sh, then all the commands will be executed once in the established sequence. Therefore, if the same sequence of commands has to be executed later, then you will only have to execute the script this way, instead of typing everything, command by command.

Exercise 2

Generation of information.

In this task, you must execute a set of linux commands for the generation of information about the created directory structure, the content of these files, and other system information

1. First, execute the command **date** in order to show the present system date.
2. Second, show the present directory where you are located, that can be whichever in general. To do this, use the command **pwd**.
3. Show all the processes that are being executing in the Linux system in this moment. To do this, use the command **ps**.
4. Using the commands **cd** and **ls**, show the information about the files and subdirectories that are under each one of the 8 folders of the working group. The information must show at least the file or/and directory creation date, their size, and permissions.

5. Using the command **grep**, show all the files that contain the string “Project”, that are under some directory of the structure of the working group.
6. Finally, put all the sequence of commands that you have done for the generation of information in a file called information.sh with a text editor and save it. You must write a line for each one of the commands and in the same order that you did in this exercise. Futhermore, add the following text in the first line of the file:

   ```
   #!/bin/bash
   ```

   Assign permissions for the execution of the new created file.

Exercise 3

Removal of folders and files.

In this task you must execute a set of Linux command for the partial removal of the directory structure, as well as some files

1. Remove completely all the files and directories that are under the directory Projects, including the same Project directory. You must do it using only the command **rm**
2. Next, remove the file that is under the folder Others using the command **rm**.
3. Next, remove the directory Others using the command **rmdir**. Note that this is possible because there is not already any files under the directory Others.

   Finally, put all the sequence of commands that you have done for the generation of information in a file called removal.sh with a text editor and save it. You must write a line for each one of the commands and in the same order that you did in this exercise. Furthermore, add the following text in the first line of the file:

   ```
   #!/bin/bash
   ```

   Assign permissions for the execution of the new created file.

Exercise 4

1. Create the directory: Lab01
2. Inside the directory "Lab01" create the file: hello.c with the following code:

   ```
   #include <stdio.h>

   int main () {

       printf ("Hello World \ n");

       return 0;

   }
   ```

3. Compile your program:

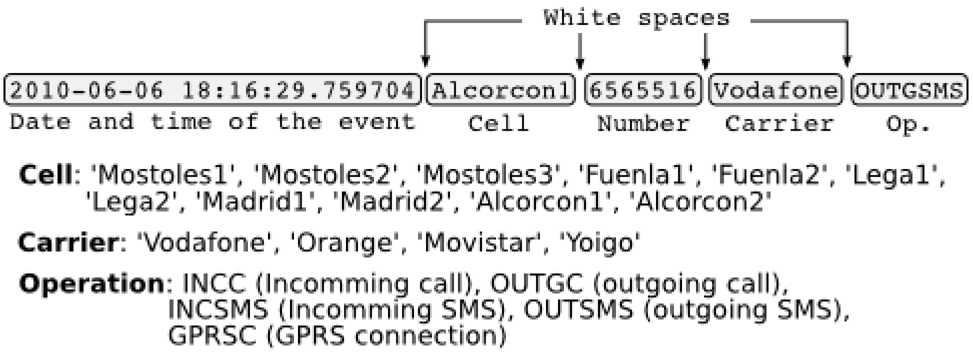
   ```
   gcc -o hello hello.c
   ```

4. Execute your program:

   ```
   ./hello
   ```

Exercise 5

The file: CellLog_1.gz has the following structure:



Cell: 'Mostoles1', 'Mostoles2', 'Mostoles3', 'Fuenla1', 'Fuenla2', 'Lega1', 'Lega2', 'Madrid1', 'Madrid2', 'Alcorcon1', 'Alcorcon2'

Carrier: 'Vodafone', 'Orange', 'Movistar', 'Yoigo'

Operation: INCC (Incomming call), OUTGC (outgoing call), INCSMS (Incomming SMS), OUTSMS (outgoing SMS), GPRSC (GPRS connection)

You have to process this file using Linux commands to obtain detailed information.

1. Decompress the file (read about the **gunzip** command).
2. Show its content in the screen (read about the **cat** and **more** commands).
3. Count the number of lines (thus the number of events) contained in the file (read about the **wc** command).
4. Select from the file only those lines that have a given word (read about the **grep** command).
5. We want to generate a report for each cell containing only the events that occurred in it. Create as many folders as needed to store these reports.
6. If the symbol ">" is added at the end of a command followed by a file name, the produced data is stored in the file instead of shown on the screen. The file is newly created. If instead of ">" the command has ">>" followed by a file name, the content produced by the command is appended to the end of the given file.

   Using this functionality and the commands seen so far, create in each subfolder a file containing only the events that belong to that cell. Execute all the commands from the folder containing all the subfolders where the reports are stored.