MYY106 Introduction to Computer Science
2nd Lab

Terminal Commands

In this lab, we will execute commands in the Terminal, so open the Terminal application (from the Applications menu).

The Terminal is the application you will probably use more often than any other. For easy access, pin it to the left panel of your desktop so you don't have to open it each time from the applications menu.
[Show me how: 🤔]

Also open the `Files` application, but not to create files and folders. Use it only to verify that the commands you give in the terminal have been executed correctly. We will create files and directories exclusively with terminal commands.

If you didn't do this in the first lab, create the course directory `myy106` (with the command `mkdir`). Enter the directory (with the command `cd`), then create a new directory named `lab2` (again with `mkdir`). Next, go to the `lab2` directory (`cd` command), and run a command that opens the text editor `pluma` and creates the file `answers-lab2.txt` 🤔. In this file you will record your answers to the questions below, where required.

First, write your full name and student ID in the text editor (or the username of your temporary account, if you don't have an ID yet). Save the file (`Ctrl+S`).

You may have noticed we sometimes use the word "**folder**" and sometimes "**directory**". That's intentional: in the graphical interface we create "**folders**" (in the Files application, for example, the command is called «Create Folder») while in the Terminal we work with "**directories**", because we use commands like `cd` (change directory) and `mkdir` (make directory). So we'll say "**folder**" for the GUI, and "**directory**" for the Terminal.

While **pluma** is open, look at your terminal: you won't see the **prompt** for a new command because the previous command (`$ pluma answers-lab2.txt`) is not yet finished. The **prompt** will reappear when you close the **pluma** window. While it remains open, you can't type other commands.

However, there is a way to get the **prompt** immediately...

By adding the symbol `&` at the end of a command, it will run in the background and the terminal won't wait for it to finish. This way the `prompt` appears at once so you can enter your next command. 🤔

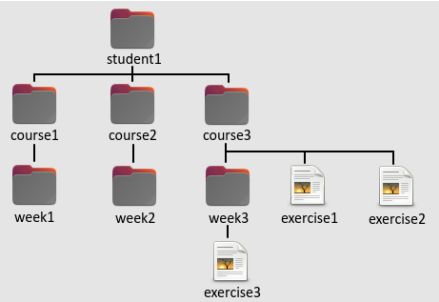
Let's now give some commands and copy our answers in the file `answers-lab2.txt`

1. In the directory `myy106/lab2`, create the hierarchical structure of directories and files shown in Figure 1, using the commands `cd` and `mkdir`. Create the files `exercise1`, `exercise2`, `exercise3` using the command `touch <filename>` in the correct directory each time. The `touch` command creates an empty file with the name you give. If the file exists, it updates the modification time. Record in the answers file the commands you used.creates an empty file with the name you give. If the file exists, it updates its modification time. Copy the commands you used to the answers file.

2. **Copy:** Using the appropriate form of the command `cp` or `cp -r`, carry out the following copy tasks of files and directories and check the result using `ls -l` (also confirm in the `Files` app). Each time, copy the commands you used to the answers file.

   α′. From the course website, download the file `lab1.pdf` with the worksheet of the first lab, and copy it into the `lab2` directory, with the new name `lab1copy.pdf`. Your browser usually downloads files into the directory (`~/Downloads`).

   β′. Copy the file `lab1copy.pdf` into the `student1` **directory**.



Σχήμα 1: Create these folders and files using terminal commands.

γ΄. Try to copy the file `lab1copy.pdf` into the `/` (root) directory. Why does the operating system complain? Record the error message in the answers file.

δ΄. Copy the files `passwd` and `group` from `/etc` into `student1/course1` directory.

ε΄. Copy the file `group` from `student1/course1` directory, into `student1/course2/ week2` directory with the new name `system_group`.

στ΄. Copy the file `exercise3` from `week3` directory into `course3` directory.

ζ΄. Copy the file `exercise3` from `course3` directory into the `student1/course1/week1` directory with the new name `week1_exercise3`.

η΄. Copy the directory `week3` into the directory `student1/course2/week2`.

θ΄. Copy all the contents (files and directories) of the directory `course3` into the directory `student1/course2`.

ι΄. Copy the directory `student1` into the directory `student2` (create it if it doesn't exist) under `myy106/lab2`.

3. **Move/rename:** Create a new directory under `myy106/lab2` named `testdir` and use the command `mv` to perform the following tasks. Verify the results with `ls -l`. Copy the commands you used in the answers file.

   α΄. Rename the directory `student2` to `tmpstudent`.

   β΄. Move all the contents of the directory `week1` from `student1/course1` into the directory `testdir` under `myy106/lab2`.

   γ΄. Move the directory `course3` from `student1` to `myy106/lab2`.

4. **Delete:** Use the commands `rm` and `rmdir` to perform the following tasks. Copy the commands you used in the answers file.

   α΄. Delete the file `lab1copy.pdf` from the directory `student1`.

   β΄. Delete the file named `exercise1` from the directory `course3` under `myy106/lab2`.

   γ΄. Delete the directory `course2` under `student1`. Why does the operating system complain? What do you observe? Find a way to perform the deletion using appropriate flags.

   δ΄. Try to delete the contents of the directory `testdir` using the `rm` command.

   ε΄. Delete all the contents of the directory `tmpstudent`.

   στ΄. Delete all the contents of the directory `course3/week3` and then delete the directory `week3` itself.

5. Check that you have answered all the questions by giving the command `tree` to view the directory tree of `~/myy106/lab2`. Copy the command's output in your answers file.

Compiling and running programs

Now that we've become somewhat familiar with executing commands in the terminal, it's time to run a program we have created ourselves. In the following activity, we will write the source code for a simple program in the C programming language, as presented in the course lectures. To be executed by the computer, the code must be converted from the C programming language to machine language, so that it can be understood by the computer. This will be done by following three basic steps: first we will compile the source code to produce assembly language, then we will use the assembler to produce an object file (machine code, but not yet executable), and finally we will use the linker to produce the executable file. For all of these steps we will use the `gcc` command with different parameters corresponding to the compiler, assembler, and linker.

α′. In the `pluma` text editor, create a new file that will contain the source code for our program.

β′. Copy the code below into the file and save it in the directory `myy106/lab2` with the name `hello-world.c`

```
#include <stdio.h>
int main(void)
{
    printf("hello, world\n");
}
```

γ′. In the terminal, run `gcc -S hello-world.c` to invoke the compiler and produce the file `hello-world.s` which contains the assembly code. Open it in `pluma` to view its contents and copy them into `answers-lab2.txt`.

δ′. Run `gcc -c hello-world.s` to invoke the assembler and produce the object file `hello-world.o` which contains machine code. Open it in `pluma` to view its contents.
Why does the text editor complain? 🤔

ε′. Run `gcc hello-world.o` to invoke the linker and produce the executable file `a.out`.

στ′. To execute your program, run `./a.out`. Copy the program's output into `answers-lab2.txt`.

What does the `./` mean before the file name?

In many commands we see a file **path**, which is the **route** we follow to reach a file, starting from the current directory. For example, the path to the file `exercise3` in Figure 1, is `student1/course3/week3/exercise3`, while we are in `myy106/lab2` directory.

We have also seen paths that begin with certain **symbols**, which point to specific directories:

~	Your home directory
/	The system's root directory (start of the disk)
.	The current directory (where you are right now)
..	The parent directory (one level above the current directory)

Submission of answers: Click the link below to open the submission form. **Copy** all the contents of `answers-lab2.txt` and **paste** them into the corresponding field (**Ctrl+A** to select all).
https://forms.office.com/e/HNRXQJmv0n

Changing Password

With your lab account, you can log in to any lab computer. However, the **username** and **password** are not stored on each computer separately, but on a server named "`scylla.cse.uoi.gr`". To change your password, you need to connect to that server.

Follow the steps below to change your account password:

α′. Connect to the server `scylla` by typing in the terminal:

```
ssh scylla.cse.uoi.gr
```

(you will be prompted to type your password)

β′. Next, use the command `passwd` to change your password.

γ′. To disconnect from `scylla` type `exit` (or press `Ctrl+d`).

δ′. If you are on a department computer, you are already in the `cse.uoi.gr` domain, so you don't need to type the full name; just type:

```
ssh scylla
```

Remote Connection

You will often need to connect to your account from a remote computer, e.g. from home.

α′. If you are working on Linux or Mac, use the `ssh` command as shown above. You must include the full address of the server: `ssh scylla.cse.uoi.gr`

β′. The username on your computer is probably different from the one you want to log in with. To log in with a different username, use the following command:

`ssh cs221234@scylla.cse.uoi.gr` ή `ssh scylla.cse.uoi.gr -l cs221234`

(of course, replace it with your own username)

γ′. The same command applies for Windows, but there is no Terminal application there. The equivalent application is called `PowerShell`. In older Windows versions which do not recognize the `ssh` command, you can download `PuTTY`, a free and open-source application that supports the `SSH` protocol.