



C1- Unix & C Lab Seminar

pool_c_d01

Day 01

Unix Environment



Instructions

Before You Go Further

- Turn in directory **pool_c_d01**.
- The exercises have to be done alone.
- Sources must be turned in with BLIH and GIT.



Exercise 1

Intranet

Connect to the Epitech Intranet using the login and password given to you, then enroll to C Pool activities.

Now connect to your mailbox and read the confirmation e-mails you recieved.



intra.epitech.eu
Office 365



If you are not register on the intranet you will **NOT** be marked.

Exercise 2

Console

All exercises of this pool will take place in the console (or "terminal").

A terminal does not have any graphical interface. Everything is done from command line.

The 'man' command, for example, can provide more information about other commands.

Use the commands 'ls' and 'cd' to show the files in the current directory and to move from one directory to another.



man ls; man cd



Exercise 3

Create directory

To help organize yourself you are going to create a directory named "Rendu". The idea is to put all your projects that you will be doing in that repository so that it is easier for us and for you to find your work.

Exercise 4

Git

To turn in your exercises, you have to use the version control software 'git'. To do this, you must first install it on your computer.

Note that to install software on your computer you must run the software installation command as administrator.



man sudo;man apt-get;
Documentation Git



Exercise 5

Blih

To turn in your exercises, you also need the **blih** script. It will let you create your "turn in" directories. You can find blih on this [link](#). Take the last version available.



The file is compressed. You can decompress it with the command tar or gunzip

The command "mv" move files and rename them. Move blih.py in the folder /usr/bin and rename it in "blih"



man gunzip; man tar; man mv; The folder /usr/bin belongs to root. To modify something inside you have to invoke to super user. man sudo;



If 'blih' does not run properly, you have to edit the shebang at the top of the file and replace by the right version of 'python'.
python --version; google shebang

Exercise 6

IDE

Create a file named "test" in your home directory and write "Les Entrechats" inside. You can display the content of the file with the command cat.

There are many ways to write in a file, but this time, you have to use the Emacs text editor.



~ represent the path to your home directory. man touch;



Exercise 7

Repository

Using blih, create a new repository called "pool_c_gecko" and give the ACL reading permission to the user "gecko".



Read the blih documentation

Exercise 8

SSH Key

Thanks to blih you just created your first git repository. This folder exists on the correction server and you must now clone a copy on your computer.

In order to communicate with the Epitech server and for this server to accept your connection, you must have a ssh key.

In your home, go to the `.ssh` directory. Then generate a ssh key called "id_rsa_coding".



`man mkdir; man ssh-keygen;`



Exercise 9

SSH Key Upload

To communicate with the server, you have to send your public ssh key to the server using blih.

Once you have achieved this step, it is still necessary to tell your machine which key to use with which server. You have to create a file `~/.ssh/config` and write the right configuration. (This file must have the rights 600).



man chmod;

All the instructions concerning sending of the key and the content of the config file are in the documentation given in exercise 6.

Exercise 10

Cloning

If the previous exercises have been properly done, you can now retrieve the folder you created on the server using git.

The repository address is: `git@git.epitech.eu:/login_x/repository_name`

In case of succes git should give you the following message :

```
Terminal
~/pool_c_d01>
[...]
```

warning: You appear to have cloned an empty repository.

```
[...]
```



man git-clone;



If git tells you that you don't have the rights, wait 5 minutes for the key to be uploaded to the server. If it still doesn't work, redo exercises 7 and 8.



Exercise 11

Turn-in (5pts)

Using `bliu`, create a new repository called `"pool_c_d01"`.

In order for your work to be marked, you must enable it to be read (reading permission). You can find the name of the user in the documentation "How to turn-in". Don't forget to also add the read rights to the user `gecko`. You will always need to give the reading rights to those two users on each of your repositories.



Do not forget to add the reading rights to each day's repository, else you WON'T be marked.

Exercise 12

Copy

Copy the file `"test"` you created in exercise 6 inside the turn in directory `"pool_c_d01"`.



`man cp;`



Exercise 13

Add

For the following exercises, you have to use git to send the file "test" on the server.

First, you have to add this file to the local git repository to enable it to be tracked.
This step must be done only one time for each new file you wish to add on the git repository.



```
man git-add;
```

Exercise 14

Commit

Once your file added, you must save the changes locally using the command "commit" of git.
Remember to always put a suitable commit message to be able to easily find an older version of your work.
In our case, you can write "Added test file".



```
man git-commit;
```



Exercise 15

Push (10pts)

Now that your changes have been saved locally, you must go through one last step. You have to synchronize your local repository with the one on the server for your modification to be accessible from others who have the correct permissions.

This step is done with the command "push" of git



man git-push;

Exercise 16

Z (5pts)

Turn in: pool_c_d01/ex_01/my_z

In your git repository, make another folder "ex_01". Then create a file called "my_z" in which you will write the letter 'z' followed by a newline.

```
Terminal
~/pool_c_d01> cat -e my_z
z$
```



Use Emacs.



Do not forget to turn in! This will not be reminded again.



Bonus

(2pts)

Turn in: pool_c_d01/ex_02/my_init_repo.sh

Write a script shell which will take the name of a repository as its first parameter and automate all the actions to initiate a repository that you have done today (creating the repository with `bliu`, and give the correct rights to the correct users). After that your script will clone the repository in your current folder. If an error would occur during the execution of a command, the script should stop right after the error without printing anything.

You do not have to handle the password encryption. The program will ask you several times.

After it is done and fully functional, you may, if you wish, place your script in a folder that is specified in your `PATH` variable so it can be used in an easier way, or just add an alias in your shell configuration file. (You will first need to look what is your `PATH` variable and how it's used by your shell or how to add an alias). Regardless of your choice you will also need to change the rights of your file so it can be executed, it is in tomorrow's notions, so try to get ahead by yourself!

Don't forget to check the probable errors that your script could have (number of arguments, if the result of your commands is valid, etc ...).

Finished? Good! Did you really check everything properly? Yes? Do it again, just to be on the safer side =D. After that if you still have some free time take a bit of time to learn more about GNU/Linux and some of the basics commands, shell configuration and personalization.



man env; see .bashrc.