Practical work for lesson 2 of course Basic tools

This list of exercises is designed only to help you learn how to work on Linux. So, if you have any questions about the exercises (or you need some help), please **do not hesitate to ask me!**

Please make a short report of used commands (via copy-paste) and short answers (if required) for each exercise and sent to my email victoriya.kashtanova@inria.fr in the end of session.

Important note: in the Linux terminal use **Shift+Ctrl+C** for copy, the action **Ctrl+C** is reserved to interrupt the processes.

Exercises

Basic tools :

1. Create directories named **lesson_1**.
2. Move all your directories and files (left from the last lecture) into the directory **lesson_1**.
3. Create directories named **lesson_2** and go there.
4. Create file **file.1.1.1** and add 3 lines with word **"Hello",** 2 lines with words **"world"** and one line with symbol **"*"**.
5. Display **file.1.1.1** content using **cat** and **less**.
6. Try **tree ~/** command. What does it do?
7. Write the directory tree of **/etc** directory to the file **tree_etc.**
8. Display the contents of file **tree_etc** page-by-page.

Zip files :

9. Install **emacs** and **zip** applications to your Linux..
   Emacs is just another test redactor for Linux (like vi/vim).
10. Download the zip file with all French zip codes from **http://download.geonames.org/export/zip/FR.zip,** unzip all content into the current directory and delete this zip-file.
11. Inspect unzipped file content with less. What is interesting in the files?
12. Search for cities with zip code 01000 in FR.txt, what are its names?
13. Search all ZIP codes found in Nice city in FR.txt, how many did you find?

Files administration:

14. Rename file FR.txt to french_zipcodes.txt.
15. Check permissions (rights) for file french_zipcodes.txt.
16. For each of the following exercises, give the corresponding chmod command with the change of permissions in symbolic format. Sequentially give the necessary permissions for the file french_zipcodes.txt so that you can :

a) Read, edit and execute your file.
b) Read, edit but not execute your file.
c) Read but not edit or execute your file.

17. Now grant all permissions for the file french_zipcodes.txt to the owner, reading only for the group and nothing for others.
18. Set the necessary permissions so that a user from your user group can read, edit, but not delete file french_zipcodes.txt. What should you do so that he can delete the file?

Shell script :

19. Create a variable named HELLO which contains "Hello, World !". Display its contents into terminal.
20. Create a variable named LONG_FILES_LIST which contains the long list of files in the current directory and use it for your directory lesson_1.
21. Use command read to ask and store a word from user. Put this word in a file user_reply.
22. Create your first Shell script hello.sh which ask user name and shows the message “Hello ‘user name’ !”. Execute your script.
23. Create a program msg_err.sh which writes a first "Normal Message" message to the terminal and a second "Error Message" message to the file which will be given in the shell input. Make sure your program works right.
24. Create a program copy.sh which has 3 argument in input and takes file (first argument), renames it with name from second argument and copy it to the directory which is the third argument. And also displays all information about the name of the program, the number of arguments, the set arguments that have been passed to this script, like here :

```
vkash@DESKTOP-3CT1O9E:~/lesson_2$ ./copy.sh tree_etc some_tree ../lesson_1/
Name of this program ./copy.sh
Nb of the arguments is 3
Arguments from input are tree_etc some_tree ../lesson_1/
```

And make the same test.
25. Using For loop create 50 files with names from file50 to file99.
26. Delete all 50 files without any loop function.
27. Create 3 programs:
    a) First program file_maker.sh to create particular number of files (first argument) with particular name (second argument + ‘_number’) in the particular folder (third argument);
    b) Second program file_renamer.sh to rename particular number of files (first argument) in the particular folder (second argument) to “name_of_file” to “name_of_file.txt”;
    c) Third program file_mover to move all .txt files from the first particular directory (first argument) to the second (second argument).