TD2/9: Fundamental Linux functionalities

Exercise 1: Access general computer informations

1. Put system up to date
2. Display
   — Linux version
   — Current processes and memory usage associated
   — Display it in a more pleasant way ("more readable for humans")
   — Number of processors
   — L1, L2 and L3 cache size
   — Disk space
   — Monted devices
   — Connected usb devices
   — Hostname

Exercise 2: Shell - Variables and scripts scope

1. Create a variable x and assign it the short text *piri pimpin*
2. Display the value of this variable
3. Add to this value the following text *piri pimpon*
   It should contain the following : *piri pimpim piri pimpon*
4. Create a folder named *my_programs*, then enter into that folder
5. Create a script named pilou that displays *pilou pilou*
6. Run this script
7. Make this script executable
8. Run the script by writting its name only
9. Programs called from the terminal are usually found thanks to a variable named *PATH*
   Display the content of the variable *PATH*
10. Add the path of your current location to the global variable PATH
11. When you are sure of the result, export it
12. Go to your home directory
13. Run your script by writting its name only
14. Change the value of the PATH in the .profile file in order to make it permanent
15. Create a new shell and run your script using its name only

Exercise 3: Scheduling task - daemon

1. Create a script *say_hello.sh*
   — Make it write the current date and time followed by *'Hello'*

— It should write it in a file named *'hellos.txt'*
— Each new output should be appened to the file (it shouldn't remove previous hellos)

2. Make the script executable
3. Use crontab to schedule the running of the script every minute

```
# Example hellos.txt content
Sat Aug  6 06:39:01 UTC 2022 - Hello
Sat Aug  6 06:40:01 UTC 2022 - Hello
```

Exercise 4: Hashing

1. Create a folder named *hash_checksum*. Go into this folder
2. Inside this folder, create two files named *.sensible_addresses* and *.sensible_passwords*
3. Display the list of files of the folder
4. Still inside the folder *hash_checksum*, create a script named *gentle_script.sh*. This script should display the following text *"Have a good day"*
5. Run the script
6. Compute the **sha256sum** of *gentle_script*. Store it into a file named *log_sha*
7. Now corrupt the file by adding a line of code that deletes any file starting with : *".sensible"*
8. Compute again the **sha256sum** of *gentle_script*. Store it into the *log_sha* file
9. Run the script
10. Display again the list of files of the folder
11. Display the *log_sha* content : are the hashes any different ?

Exercise 5: Compressing

1. Install the QPDF free command-line program.
   Part of this program is the **zlib-flate** command that compress and uncompress files using the **deflate** algorithm.
2. Create a directory *"compress"*, go into this directory
3. Create a first file *"hello"* whose content is *"Hello"*
4. Compute the deflate compression (level 1) of this file. Store the compressed file size into a file *log_compress*
5. Create a second file *"hello_multiple"* whose content is 1000 lines of *"Hello"*

6. Compute the deflate compression (level 1) of this file. Store the compressed file size into a file *log_compress*
7. Create a third file *"hello_mulitple_i"* whose content is 1000 lines of *"Hello i"* (i varying from 1 to 100)
8. Compute the deflate compression (level 1) of this third file. Store the compressed file size into *log_compress*
9. Display the content of *log_compress*
10. Compute the compression ratio of each file, also display it as a simple fraction (e.g. 12.6 => 10 :1)
11. Analyse the results

Exercise 6: ACLs : Access Control Lists

1. Create users
   — Create a user named *client_1* with password *passwd-client_1*
   — Create two other users named *contributor_1* and *contributor_2* with respective passwords *passwd-contributor_1* and *passwd-contributor_2*
2. Create groups
   — clients
   — contributors
3. Add users to their respective group
4. Check the users and groups have been successfully created
5. Create a folder *lika_project* and give it the following authorizations to groups
   — clients : read
   — contributors : read and write
6. Also use the command **ls -l** and notice the change on *lika_project* folder
7. Change user and become as a client, then try deleting the folder
8. Now change user and become as a contributor, then try deleting the folder
9. Check who is the current user