Introduction to Shell

Nasser-Eddine MONIR

October 9th, 2024

Practical-1

UNIVERSITÉ DE LORRAINE

UL-IDMC Introduction to Shell

nasser-eddine.monir@inria.fr nasseredd.github.io

Institut des sciences du Digital Management & Cognition

COMPOSANTE DE L'UNIVERSITÉ DE LORRAINE

Exercise 1

File Manipulation

1. Download **Practical-1/** directory in your **Desktop/** (or **Bureau/**)
2. Change Directory to **~Desktop/Practical-1/**
3. Create a new directory called **exercise_1/**
4. Change Directory to **exercise_1/**
5. Create 3 text files : file1.txt, file2.txt, file3.txt
6. Append the text of dump_text.txt in file2.txt

Exercise 2

Text Processing

1. Create a new directory called **exercice_2/**
2. Change directory to **exercice_2/**
3. Create a text file and name it **myfile.txt**
4. Append the text of **inria.txt** in **myfile.txt**
5. Display lines that contains these terms: « Inria », « Alors », « mission ».

Exercise 3

Text Processing

1. Create a new directory called **exercice_3/**
2. Change directory to **exercice_3/**
3. Create a text file and name it **file1.txt**
4. Append the text of **university_of_lorraine.txt** in **file1.txt**
5. Considering the **wc** command (check the doc), answer to the following questions:
   - How many lines ?
   - How many characters ?
   - How many words ?

Exercise 4

Sorting

1. Create a new directory called **exercice_4/**
2. Change directory to **exercice_4/**
3. Copy the file **fruits_vegetables.txt** in your current folder
4. Sort the list of fruits and vegetables alphabetically
5. Sort the list of fruits and vegetables alphabetically in descending order
6. Sort the list of fruits and vegetables with unique words

Exercise 5

Bash Scripting

1. Create a new directory **exercise_5**/ and change directory to it
2. Create a new file and name it **my_script.sh**
3. Open the file with native or non-native editor
4. Write a bash script to say « Hello World »
5. Make the script executable (giving the right)
6. Run the script

Exercise 5 (Bonus)

Bash Scripting

Choose one of the three first exercises, and write a shell script that corresponds to all the instructions.

Exercise 6

Bash Scripting

Write a Bash script that :

1. Prompts the user to enter a number
2. Use an if-else condition to check if the number is even or odd
3. Display an appropriate message based on the result

Exercise 7
Bash Scripting

Write a script that simulates user authentication :

1. Hardcode a username and password in the script (ground truth)
2. Prompt the user to enter a username and password
3. Use if-else conditions to verify if the entered credentials match the hardcoded ones
4. Provide access if the credentials are correct and deny access if they are not. (echo “Authentication successful. Welcome, « username » !”)

Exercise 8

Bash Scripting

Write a script that lists all files and directories in the current directory :

1. Use a loop to iterate through the items and display their names
2. Differentiate between files and directories in the listing

Exercise 9

Bash Scripting

Write a script that :

1. Create a directory named **exercise_9**.
2. In **exercise_9**, generate 100 **.txt** files inside the **text_files** subdirectory, with random number filenames between 10 and 10,000.
3. Move all the odd-numbered **.txt** files to the **French/** directory and the even-numbered ones to the **English/** directory.
4. Count and display how many files were moved to each subdirectory
5. In **English/** directory, create a subdirectory named **old/**
6. Copy all the files from **English/** to **old/** whose filenames are divisible by 5.

Exercise 10 (Bonus)
Training a (fake) ML model

You want the user to provide a stage number (between 0 and 3) when running the script. If the user does not provide any input, you should use a default stage.

You should set a default stage to be used when the user doesn't enter any stage number.

Based on the stage number provided, you need to run a corresponding Python script. For example:

- Stage 0: load_dataset.py
- Stage 1: parsing_data.py
- Stage 2: train.py
- Stage 3: eval.py

If the user provides a stage number that is not between 0 and 3, the script should display an error message and exit.
After running the script for the specified stage, the script should continue running the next stages in sequence (i.e., if the user starts at stage 1, stages 2 and 3 should be executed automatically afterward).
After each script execution, the stage should be incremented by 1 so that the next stage's script can be run.
After writing the script, you need to ensure it runs correctly from any given stage (0, 1, 2, or 3) and continues until all stages are completed.