Assisgnment I: Numerical and Computational Methods in Reseach (PYL 800) 2022-2023

Course Instructor: Suprit Singh

Working with the terminal in Linux

Q1. Here is an exercise that employs the Linux command **ls**:

1. Open a terminal window on your Linux machine.
2. Navigate to your home directory by running the command **cd ~**
3. Run the command **ls** to see a list of the files and directories in your home directory.
4. Now create a directory named "test_dir" using command **mkdir test_dir**
5. Now navigate inside the directory using command **cd test_dir**
6. Now create a few files inside the directory using command **touch file1 file2 file3**
7. Now navigate back to your home directory by running the command **cd ~**
8. Now run the command **ls -l test_dir** to see the details of files and directories in the "test_dir" directory
9. Now delete the directory test_dir using command **rmdir test_dir** or **rm -r test_dir**
10. Run the command **ls** again to confirm that the "test_dir" directory has been deleted.

Note: This exercise creates some files and a directory on your machine and then it deletes them so, be careful when running this commands.

Q2. Here is an exercise that employs the Linux commands **echo**, **cat**, and **sort**:

1. Open a terminal window on your Linux machine.
2. Create a new file named "numbers.txt" using the command **echo "3" > numbers.txt**. This command will create a new file named "numbers.txt" and insert the number "3" as its first line.
3. Append more numbers to the file by running the command **echo "1" >> numbers.txt** and **echo "4" >> numbers.txt** and **echo "2" >> numbers.txt**
4. Use the command **cat numbers.txt** to view the contents of the "numbers.txt" file, you should see a list of numbers "3 1 4 2"
5. Use the command **sort numbers.txt > sorted_numbers.txt** . This will sort the contents of the "numbers.txt" file and write the sorted output to a

new file named “sorted_numbers.txt”
6. Use the command **cat sorted_numbers.txt** to view the contents of the “sorted_numbers.txt” file, you should see the same numbers but in sorted order “1 2 3 4”
7. Use the command **sort -r sorted_numbers.txt > reverse_sorted_numbers.txt** . This will sort the contents of the “sorted_numbers.txt” in reverse order and write the output to a new file named “reverse_sorted_numbers.txt”
8. Use the command **cat reverse_sorted_numbers.txt** to view the contents of the “reverse_sorted_numbers.txt” file, you should see the numbers in reverse order “4 3 2 1”
9. clean up the created file by running the command rm numbers.txt sorted_numbers.txt reverse_sorted_numbers.txt

This exercise shows how to create a file using **echo**, view the contents of a file using **cat**, sort the contents of a file using **sort**, and how to sort in reverse order by using flag -r.

Q3. Here is an exercise that employs the Linux command **unzip**:

1. Open a terminal window on your Linux machine.
2. Navigate to the directory where you want to unzip the file. For example, if you want to unzip a file in the Documents directory, you can navigate to that directory by running the command **cd ~/Documents**
3. Download an example zip file by running the command wget https://github.com/openai/gpt-3-examples/raw/master/image-captioning/images.zip
4. Use the command **ls** to verify that the zip file is in the current directory.
5. Use the command **unzip images.zip** to extract the contents of the zip file into the current directory.
6. Use the command **ls** again to see the contents of the current directory, you should now see all the files and directories that were inside the zip file
7. Use the command **unzip -l images.zip** to list the contents of the zip file, it will display the file names and the compressed size
8. Use the command **unzip -d ~/Downloads/images images.zip** to extract the files to a specific directory, in this case the directory name is images and it is located inside the Download folder
9. Use the command **rm images.zip** to remove the zip file

This exercise shows how to download a zip file using **wget**, extract the contents of a zip file using **unzip**, how to list the contents of a zip file using **unzip -l**, and how to extract the contents of a zip file to a specific directory using **unzip -d** .

Q4. Here is an exercise that employs Linux command-line tools to work with a CSV file:

1. Open a terminal window on your Linux machine.
2. Navigate to the directory where you want to create the CSV file. For example, if you want to create the file in the Documents directory, you can navigate to that directory by running the command **cd ~/Documents**
3. Create a new file named "data.csv" using the command **touch data.csv**.
4. Populate the file with sample data using the command echo "Name,Age,Gender" >> data.csv && echo "John,20,Male" >> data.csv && echo "Jessica,25,Female" >> data.csv && echo "Michael,30,Male" >> data.csv
5. Now use the command **cat data.csv** to view the contents of the "data.csv" file, you should see the data inside the file
6. Use the command **cut -d "," -f 1 data.csv** to extract the first column of the CSV file, which in this case is the "Name" field
7. Use the command **grep "^M" data.csv** to find all the rows that start with 'M' in the "Name" field
8. Use the command **awk -F "," '$2 > 25 { print $1 }' data.csv** to extract the "Name" field of the people older than 25
9. Use the command **sort -t "," -k 2 -n data.csv** to sort the contents of the CSV file by the second field (Age) in ascending order
10. Use the command **rm data.csv** to remove the CSV file when you're done working with it

This exercise shows how to create a file using **touch**, populate data inside a file using **echo** and **&&** operator, how to view the contents of a file using **cat**, how to extract a specific column of the CSV file using **cut**, how to find specific rows using **grep**, how to extract specific fields of the CSV file based on certain conditions using **awk**, and how to sort the contents of a CSV file using **sort** command, it uses -t option to specify delimiter, -k option to specify field and -n option to sort numerically.

Q5. Here is an exercise that employs the Linux command **diff**:

1. Open a terminal window on your Linux machine.
2. Navigate to the directory where you want to create the files for the exercise. For example, if you want to create the files in the Documents directory, you can navigate to that directory by running the command **cd ~/Documents**
3. Create a new file named "file1.txt" using the command **touch file1.txt**.
4. Populate the file with sample data using the command **echo "This is file1" > file1.txt**
5. Create another file named "file2.txt" using the command **touch file2.txt**.
6. Populate the file with sample data using the command **echo "This is file2" > file2.txt**
7. Now use the command **diff file1.txt file2.txt** to see the difference between the two files. The command should indicate that the files are different

8. Modify the file1.txt by running the command **echo "This is updated file1" > file1.txt**
9. Use the command **diff file1.txt file2.txt** again to see the difference, this time the command should indicate that the line 1 of the file1.txt is different from the line 1 of the file2.txt
10. Use the command **diff -u file1.txt file2.txt** to show the difference in a unified format.
11. Now create a new file named "file3.txt" using the command **touch file3.txt**
12. Populate the file with same sample data of file1.txt using the command **echo "This is updated file1" > file3.txt**
13. Use the command **diff file1.txt file3.txt** to see if the files are identical.
14. Use the command **rm file1.txt file2.txt file3.txt** to delete the files when you're done working with them.

This exercise shows how to create a file using **touch**, populate data inside a file using **echo** , how to compare the contents of two files using **diff** command and how to see the differences in unified format using **diff -u** option, it also show you how to compare two identical files.

Q6. Here is an exercise that employs Linux command-line tools to rename a bunch of files in a serial order:

1. Open a terminal window on your Linux machine.
2. Navigate to the directory where the files you want to rename are located.
3. Use the command **ls** to view the list of files in the current directory
4. Use the command **touch file{1..5}.txt** to create 5 new files named "file1.txt", "file2.txt", "file3.txt", "file4.txt", and "file5.txt"
5. Use the command **ls** to verify that the files have been created.
6. Use the command **for i in {1..5}; do mv file${i}.txt file${i}_new.txt; done** this command will rename the files in serial order in the form file1_new.txt, file2_new.txt, file3_new.txt, file4_new.txt, and file5_new.txt
7. Use the command **ls** to see the list of the files in the directory, you should see the files with new names
8. Use the command **rm file{1..5}_new.txt** to delete the files with new names

This exercise shows how to create multiple files with similar naming pattern using **touch** command and **{}**, then it uses a **for loop** with **mv** command to rename them in serial order, where the **for** loop iterates through a range of numbers, and the **mv** command renames the files with a new prefix at each iteration. This is a powerful command to rename a bunch of files in a directory by providing a simple numbering or iterating through a pattern that you want to use for your files.