

## Operating Systems Laboratory (CS39002)

Spring Semester 2019-2020

**Assignment 1:** Familiarization with shell scripting (commands like awk, grep, sed, sort, cat)

**Assignment given on:** January 13, 2020

**Assignment deadline:** January 20, 2020, 1:00 PM

**1.** Write programs in shell script under the Linux environment that would run in bash terminal and perform the following Tasks.

### 1.a. Simple calculator (5 marks)

- Take in three arguments: first is an integer, separated by a space second argument is any one of the following operators: addition (+), Subtraction (-), Multiplication (\*), Division (/) and Modulo (%), third argument is again an integer. Compute and print the expression created by argument 1, argument 2, argument 3.
- Name your shell script "Assgn1\_1a\_< groupno>\_<roll no. 1>\_< roll no. 2>.sh"
- Example input: Assgn1\_1a\_< groupno>\_<roll no. 1>\_< roll no. 2>.sh 1 + 2
- Example output: 3

### 1.b. Adding line numbers and headers (5 marks)

- You have a text file with around 500,000 lines (download from [https://cse.iitkgp.ac.in/~mainack/OS/assignments/01/1b\\_input.txt](https://cse.iitkgp.ac.in/~mainack/OS/assignments/01/1b_input.txt)):
- Write a shell script which will read this file and add line numbers to the beginning of each line. These line numbers should be followed by spaces. Also add the following headers in the two columns of the new file: "Serial", "Random string"
- Name your shell script "Assgn1\_1b\_< groupno>\_<roll no. 1>\_< roll no. 2>.sh"

### 1.c. Finding GCD (10 marks)

- Write a shell script which will take less than 10 integers and output their GCD.
- Name your shell script "Assgn1\_1c\_< groupno>\_<roll no. 1>\_< roll no. 2>.sh"
- Example input: Assgn1\_1c\_< groupno>\_<roll no. 1>\_< roll no. 2>.sh 4,8,16,56
- Example output: 4

### 1.d. Sorting and merging large number of files (10 marks)

- You are given a folder with hundreds of text files (download from here and unzip): <https://cse.iitkgp.ac.in/~mainack/OS/assignments/01/1.d.files.zip>
- You need to sort each of these individual files in decreasing order, write the sorted files in a new "1.d.files.out" folder (keep the name same), and then merge these individual sorted files into a single sorted file (name it "1.d.out.txt") which is sorted in decreasing order. Write a shell script which will iterate over the given files and perform this task.

- Name your shell script “Assgn1\_1d\_< groupno>\_<roll no. 1>\_< roll no. 2>.sh”

### 1.e. Creating frequency distribution from a large file (15 marks)

- You have a text file with around 1.5 million lines (download from [https://cse.iitkgp.ac.in/~mainack/OS/assignments/01/1e\\_input.txt.zip](https://cse.iitkgp.ac.in/~mainack/OS/assignments/01/1e_input.txt.zip) and unzip)
- This text file has four space-separated columns, example of three lines from the file:
 

```
1 ib Jim 34
1 cr JoHn 24
1 ut MaRY 46
```
- Write a shell script which will take two inputs: (i) the input file, and (ii) a column number as a user provided input (varies from 1 to 4). For that column number the code will read entries of that column from the input file (e.g., in case of column number 3 as input the entries will be Jim, JoHn, MaRY), and convert these entries to lower case.
- Finally the script should create a file “1e\_output\_<column\_number>\_column.freq”. This file will have two columns: first is a string (converted to lowercase) from the given column and second is the frequency of that string. Sort the “1e\_output\_<column\_number>\_column.freq”. The file should be sorted in decreasing order of the second column values (i.e., frequency).
- Name your shell script “Assgn1\_1e\_< groupno>\_<roll no. 1>\_< roll no. 2>.sh”

### 1.f. Cleaning up graphs (15 marks)

- You have a large graph (around 5 million edges) as edgelist (download it from [https://cse.iitkgp.ac.in/~mainack/OS/assignments/01/1f\\_graph.edgelist.zip](https://cse.iitkgp.ac.in/~mainack/OS/assignments/01/1f_graph.edgelist.zip) and unzip).
- In the graph there are n nodes which are labeled as integers. In the file a line “n1 n2” means that there is an undirected edge between nodes labeled n1 and n2
- Now, write a shell script that will take your graph as input and remove (i) self-edges (i.e, lines like “345 345”) (ii) duplicate edges (i.e. if “2 1” exists, then “1 2” should not exist) and finally, (iii) parallel edges (i.e., there should be only unique edges in the file)
- Your script should write the new clean output graph in a file named “1f\_output\_graph.edgelist”
- Finally your code should then parse the new “1f\_output\_graph.edgelist” and print out 5 nodes which have highest undirected degree and their corresponding undirected degrees. Each node should be printed on a separate line as follows:
 

```
<node_with_higest_degree> highest_degree_in_the_graph
<node_with_second_higest_degree> second_highest_degree_in_the_graph
...
...
```
- Name your shell script “Assgn1\_1f\_< groupno>\_<roll no. 1>\_< roll no. 2>.sh”

### Submission Guidelines:

- Create a single zip file with all the shell scripts (with specified names). Name the zip file : “Assgn1\_< groupno>\_<roll no. 1>\_< roll no. 2>.submission.zip”

- You must show the running version of the program(s) to your assigned TA during the lab hours.