

Operating Systems Laboratory (CS39002)

Spring Semester 2020-2021

Assignment 1: Familiarization with shell scripting and shell commands

Assignment given on: January 14, 2020

Assignment deadline: January 21, 2020, 1:00 PM

#READ THE FULL INSTRUCTIONS VERY CAREFULLY FIRST

0. Overview

In this assignment we will start with a simple exercise – familiarization with shell scripting. Shell or “bash” (a variant of shell software) is your interface to access a multitude of software (often called “commands”), some of which are directly provided by OS. shell OR command line OR terminal can be simply accessed in your Linux distribution by opening terminal app. Search Google to know more.

0.1. Shell scripting example

Let’s take an example: You have a file named “tmp.txt” which contains multiple lines, each line is a number. You want to numerically sort those numbers. So, you can simply

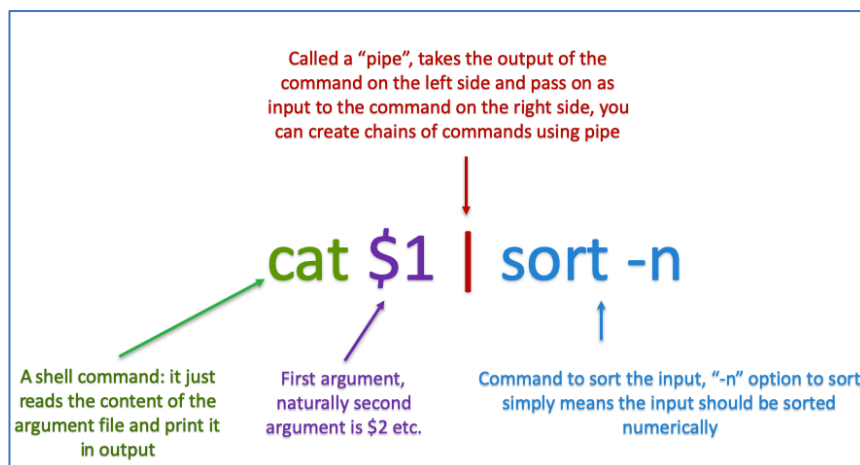
- Open up terminal.
- Type `cat tmp.txt | sort -n` and press Enter.
- The sorted number will be printed in the terminal

You can also store it in a file for permanent use.

- You open a file called “sort.sh” where you would store your bash script.
- You want sort.sh to take filename as input
- So, within sort.sh you would put `cat $1 | sort -n`, save the file and run the following command: `sh sort.sh tmp.txt`
- sort.sh is called a shell script

0.2. What is going on?

Here is a dissection of the script:



Of course, like any self-respecting programming language you can do everything in shell script (loops, conditionals, case-switch etc.). However, the power of the shell script is derived from clever use of the already-existing software and functionalities provides by shell script (e.g., cat, sort, | etc.)

0.3. Shell commands to be familiar with

There are a **few shell commands** that you should be familiar with to efficiently use shell (aside from the shell scripting syntax). Search google (also write `man <commandname>` in terminal) to know more about them (some or all of them might also be useful for you to solve this assignment). Each command can take multiple arguments to perform various tasks.

- **cat**
- **sort**
- **awk (also search awk one-liners)**
- **head**
- **tail**
- **cd**
- **mkdir**
- **ls**
- **touch**
- **file**
- **tr**
- **grep**
- **alias**
- **dd**
- **nohup**
- **wc**
- **split**
- **cut**
- **cmd1 &**
- **cmd1 | cmd2**
- **file1.txt < file2.txt**
- **file1.txt > file2.txt**

0.4. Grading

- You need to submit your assignment to Moodle within deadline.
- Make sure **the submission naming scheme exactly follows as given in the assignment.** Naturally if you mistype your group name, your marks might be awarded to another group.
- Submit **ONE assignment** per group.
- **We will award 90% marks for this assignment if your solutions to the problems are correct** and you can demo the correctness to your instructor in the lab with instructor mentioned test cases (means your code runs in *reasonable* time for those cases).
- **Rest 10% will be reserved for the top 5 groups** who can write the code (in total) using least number of words (akin to solving a maths problem in least number of steps). Here “words” will mean number of words calculated by “wc -w” command on terminal.
- **We will release a leaderboard for this assignment.**

With this introduction let's start the problems. Best of Luck!

1. Problems

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

1.a. Finding GCD (10 marks)

- Write a shell script which will take less than 10 integers (comma separated) and output their GCD.
- Name your shell script “Assgn1a_< groupno>_<roll no. 1>_< roll no. 2>.sh”
- Example input: Assgn1_1a_< groupno>_<roll no. 1>_< roll no. 2>.sh 4,8,16,56
- Example output: 4

1.b. 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/2021/01/1.b.files.zip>
- You need to sort each of these individual files in decreasing order (numerically), write the sorted files in a new “1.b.files.out” folder (keep the name same), and then merge these individual sorted files into a single sorted file (name it “1.b.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 “Assgn1b_< groupno>_<roll no. 1>_< roll no. 2>.sh”

1.c. Creating frequency distribution from a large file (10 marks)

- You have a text file with around 1.5 million lines (download from https://cse.iitkgp.ac.in/~mainack/OS/assignments/2021/01/1c_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 “1c_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_1c_< groupno>_<roll no. 1>_< roll no. 2>.sh”

1.d. Creating an extract utility (20 marks)

- Exaction of a compressed file via command line is often a cumbersome task. For example, uncompressing a file in “a.tar.bz2” requires the command “tar xvjf a.tar.bz2” and a file “a.tar” requires remembering and typing the command “tar xvf a.tar”.
- So, in this assignment write an “extract” script which will take a filename as input and extract the file in the current directory.
- Your function should support the extraction of files which are compressed using a number of protocols. For this assignment make sure that your script supports files with extensions **tar.bz2, tar.gz, bz2, rar, gz, tar, tbz2, tgz, zip, Z, 7z**
- Of course, if the data is not a known file type then extract should output “Unknown file format: cannot extract”
- Name your shell script “Assgn1d_< groupno>_<roll no. 1>_< roll no. 2>.sh”

1.e. Creating a random password generation function (20 marks)

- Often in different webservices you would need random passwords (with upper- and lower-case letters as well as numbers and “_” character).
- So, create a “genpasswd” shell script which will generate a pseudorandom password with with upper- and lower-case letters as well as numbers and “_” character with the help of input from “/dev/urandom” which is the random number generator included in your linux distribution.

- You script should take the optional length of password as input (default length 16 characters)
- Name your shell script "Assgn1e_< groupno>_<roll no. 1>_< roll no. 2>.sh"

Moodle 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.