Introduction to Software Systems

Assignment - 1 (Bash Scripting)

Due date: 2nd February, 11:55 pm

Question 1:

Store Music
(TA to contact regarding doubts in the question: Projit)

Aim:
To create a document and supporting commands to store songs/music that you listen to, so that you can look back on them when you're bored and trying to find out songs that you've liked in the past. Kind of like a song diary.

Requirements:

- Create a storage document, which holds all the details that you add. Simplest form would be some kind of delimited document, but feel free to use multiple documents or even a database to store the information.

- Create commands(name them whatever you want):
  - Add entry:
    - Takes n arguments (your wish as to what data you want to keep).
    - At least 1 (or more) mandatory argument
    - At least 1 (or more) non-essential arguments
    - Suggestion: Song name, artist name, genre, youtube link ...etc
    - Should check to ensure no duplicate
    - Suggestion: Keep an id parameter, would be helpful in identification for editing
    - Use redirection (<, >) to add to the file.

  - Edit entry
    - Allows updating the fields of a certain song entry. (ex. fixing spelling mistakes, adding in new youtube link).
    - Suggestion: Could be implemented as a Delete + Add

  - Delete entry
    - Remove a song entry from your document.
    - Deletion should be achieved in a single command. (See: sed)

- View all entries
  - Put a "nice view" to all the songs you have. (Create a table-like view?)

- View select entries
  - View all the songs satisfying a certain requirement. (ex. all songs of 1 artist, all songs of same genre)

BONUS:
- Allow for the user to add in custom fields

Submission:
- Script to setup the whole system on someone's system. Assume no sudo permissions (For your own use, you may want to move some of your scripts to /usr/local/bin so that they are accessible throughout your system, or add them to your path)
- Scripts for each of the commands(add, edit, view all, view select, and any additional commands you've included)

Question 2:

Creating your own search engine !!
(TA to contact regarding doubts in the question: Mugdha)

Problem statement: As cool as this sounds, you will be using a set of simple bash commands to count the the number of times a given string occurs in the source of a web page. Name your script "searching.sh".

Example usage:
bash searching.sh searchString http://www.example.com/routexyz
Output:
searchString 42

Your script takes 2 command line arguments: the string to look up for, and the URL of the website you want to look at.
The output contains the string to look up for, followed by its frequency of occurrence. Make sure you handle exception and corner cases. Also, handle errors gracefully. If the right number and type of arguments are not given, display an error message with the right usage instructions. Submit only one file, "searching.sh".

Question 3:

(TA to contact regarding doubts in the question: Anubhab)

A popular time management technique uses a timer to break down work into intervals (usually 25 mins) followed by a small break.
The task is to create a BASH script to aid this process. Take a command line argument for the number of iterations (work + break). Print notifications about the breaks (5 mins) or time to work (25 mins) in the terminal. Every 4 iterations, include a long break (15 mins). Print notification when all the cycles are complete.

Example usage:
bash timer.sh 2
Output structure:
#1 work
#1 break time
#2 work
#2 break time
Finished

You are free to change the content of the output to make it more meaningful as you wish but iteration number needs to be present and general structure must be the same.
Bonus:
Instead of printing to the terminal, send notifications. You are free to use your package manager to install additional non-default commands for this task (Only for the bonus and not the rest of the question).

Question 4:

Reminders
(TA to contact regarding doubts in the question: Karandeep)

Aim : To create a program that allows you to create, edit, list, delete and send reminders as desktop notifications.
Store all reminders in a delimited file(s).

Commands:
- Add reminder :
  - Take command line arguments for time, body, frequency(optional)
  - Your code should assign an id to the reminder

  - Corresponding reminder should be sent at the time entered
- List reminders :
  - With no arguments, list all sorted by time of reminder
  - Optional arguments for filtering by date, substring of body
- Edit reminder :
  - Take arguments for id and the new fields
- Delete reminder:
  - Take argument for id

Submission: One script to install all dependencies, scripts for each command

Question 5:

(TA to contact regarding doubts in these questions: Ali)

* Write a bash script which will do the following:
  * Print **recursively** the last modified date and time and the file name of the all the files in the current working directory in a given format.(Use piping)
    Example Output: If your current directory contains two files A and B. Then
    output should be as follows:
    Last_Modified_Date_of_A Last_Modified_Time_of_A Relative_path_of_A
    Last_Modified_Date_of_B Last_Modified_Time_of_B Relative_path_of_B

  * Find all the commands that starts the word “lo” and store the commands and their small descriptions in a file named your_roll_no.txt(For eg: 201801159.txt). Each line in the file should contain the command along with its small description.
  * Display the number of lines and the length of the longest line in the above created File.
  * Replace all the occurrences of the word “function” with “method” in the above file. In addition, create a backup of your original file “your_roll_no.txt”.

Submission: script1.sh, your_roll_no.txt, backup file

* Adult income census was conducted and the output of the census contains 2 csv files named file1.csv and file2.csv. Write a script named script2.sh which will perform the following operations:
  * Concatenate these 2 files into a single file named target_file.csv
  * Create a header file named header.csv which will contain:

"Age,workclass,fnlwgt,education,education-num,marital-status,occupation,relationship,race,sex,capital-gain,capital-loss,hours-per-week,native-country,class".
    Add this header file at the beginning of the target_file.csv

* The target_file.csv would contain some missing values denoted by '?', replace these missing values with your roll number.
Submission: script2.sh, target_file.csv

Question 6:

(TA to contact regarding doubts in the question: Mohit)

Instructions

- Read the question carefully
- Use the file 'movies_input' as an input for testing your script

Problem Statement :

Samyak and Archit are very good friends. Archit is a total movie freak. He has lots of movies in his laptop. Samyak, gave him a list of movies and their IMDB ratings. He was very delighted and wanted to sort all the movies he has as per their IMDB ratings & put them in different folders. Your job is to help Archit.

Input format :

Command will be of the form :
bash <your_script.sh> <movies_rating_file_path> <all_movies_folder_path>

<movies_rating_file_path>: Path to the movies_input file
<all_movies_folder_path>: Path to the directory which would contain movie files

Eg : bash ./script.sh /home/Archit/Samyak_movies_rating.txt ./media/All_Movies/

Ratings file format :

Format of input <movies_rating_file_path> :
<movie_name>:<imdb_rating>

For ex -

Simmba:6.6
Andhadhun:8.8
Stree:7.2
Sanju:7.6
Padman:8.4
Zero:6.1

.....

Note - *There is no spacing between the movie name, colon and the imdb rating.*

Your script should do this-

1). Create 4 folders at <all_movies_folder_path> : Bad, Average, Good, Awesome.
2). For all the movies present in <all_movies_folder_path>, find their imdb ratings from <movies_rating_file_path> (i.e. Using file 'movies_input'). Now create empty movie files in any one of the 4 folders according to the following rule:
If the rating < 5 : Bad
If rating is >=5 and <8 : Average
If rating >= 8 and <9 : Good
If rating >= 9 : Awesome

After running the script, the movie files should be present in their respective folders as per their rating.

Submission format-

You will have to submit one bash script file by name <Roll_no>.sh containing the bash script .

Assumptions-

- Name the empty movie files as <movie_name>.mp4
- The name of the movies will be a single word.

ALL THE BEST !!!