CSL503: Computer Systems Engineering

Assignment-1

Due Date: 31-August-2024 11:59 PM
Maximum Points: 100

Question-1:

Assume that you have a directory containing several .csv files. Write a shell script that takes a string as a command line argument and computes the sum of all occurrences of the given word in all the .csv files. You should use as many linux commands as possible.

Question-2:

Assume that you have three files CSP100.txt, CSP101.txt, CSP102.txt. Each file name denotes course name. Each file contains several lines in the below format

RollNo, Grade
RollNo has 8 digits, where each digit can be [0-9]
Grade: Can be any of A, B, C, D, E, F

Write a shell script for each of the following

1. List all the roll numbers who obtained an A grade in CSP102 but a B grade in CSP101
2. That list course name which has maximum number of F grades
3. Print the roll numbers of the students who have obtained E grade in any course. Replace all E grades with a C grade for the course, which has maximum E grades in it.
   [Hint: Use **sed** command inside the shell script]

Question-3:

Write a shell script that takes the name of a folder as command line argument, and delete all files of size 0.

Question-4:

Write a shell script that takes string1, string2, and a directory name as three command line arguments, and replaces all the occurrences of string1 to string2 in the first 10 lines of all .txt files in the given directory.

Question-5:

Write a shell script that changes all the filenames ending in **.txt** to **.dat** in the current directory.

Question-6:

Implement a shellATM machine using the shell script with the following features.

1. *Account.txt* contains several lines in the below format
   Account Name, Card Number, Email ID, Balance

   Account name: It should contain First name followed by Last name (with space separated). First and Last names start with capital letters followed by any number of lower letter alphabets.

   Card Number: It has 16 digits in the below format <4 Digits><space><4 Digits><space><4 Digits><space><4 Digits>

   Email ID: It has the format username@Domain.
   The username should start with an alphabet (lower/upper) followed by any number of alphabet or digits. Domain should use <string>.<string>.<string>, where strings contain only lower Alphabets.

   Balance: Any number >= 0.

2. *Credentials.txt* has the below format.
   Card Number, Password.

   Card number format follows as listed in *Account.txt.* Passwords can be any string.

Step-0: Your shellATM script upon execution should show a welcome screen as follows.

```
**** Welcome to My shellATM ****
**** Press any key to continue ***
```

Step-1: Once any key is pressed, the terminal screen should seek the credentials of Card Number and Password. If both the Card Number and Password match as listed in Credentials.txt, it should authenticate the users. Otherwise, it should ask to enter the details again. [Hint: You can use grep commands here]

Step-2: A successful authenticated user should get the following options.

1. Withdraw cash
2. Deposit cash
3. Settings
4. Exit

Withdraw Cash/Deposit cash options should accept a non-negative number and subtract/add the cash amount entered by the user. Once the operation is complete, your terminal should show the updated balance on the screen. *[Hint: Use grep with regular expression to validate the number and sed for updation of balance].*

Settings: Option should provide the user the user to change the email address. The new account and email address should use the same syntax as listed in the *Account.txt.* Any invalid entry should throw an error. If the new entries are valid, you should update the Account.txt. *[Hint: You can use grep for checking the patterns and sed for replacement]*

After completion of any of the options in Step-2, the terminal should go to the welcome screen.

Suggestion:

1. Your shell script should use as many linux commands as possible to minimize the work.
2. You can use **read** to get the input from the user.