Shell Scripting

Q1- *Motivation* ☺

Write shell Script to represent any amount of money by using the fewest number of coins

The script take one argument that is amount of money.

Coins available: 5, 10 and 25

Example: ***./Coins*** 115

4 of 25

1 of 10

1 of 5

Q2- Objective: understanding File permissions

Write shell script ***"resolve_perm"*** take two arguments: username ***usr*** and filename ***fname***, then resolve and show permissions of this user on this file.

Q3- Objective: Sorting Files

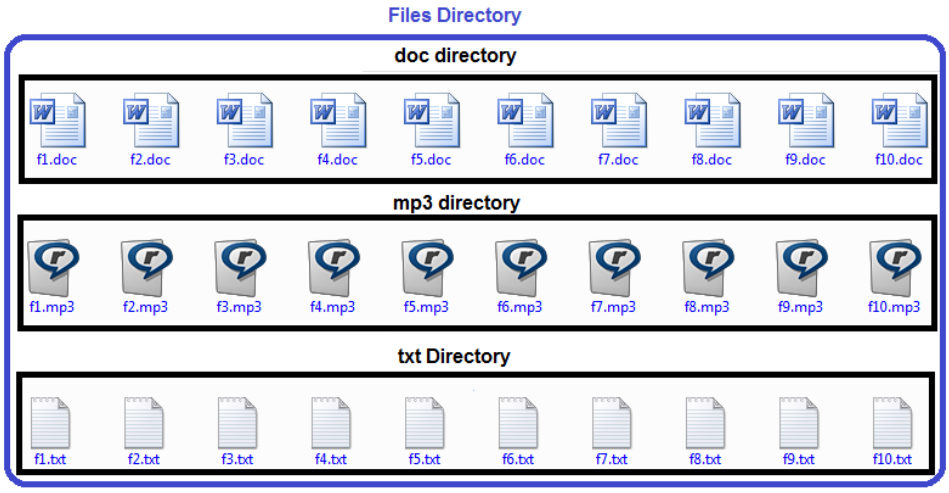
1. Write shell script ***"make_files"*** take unknown number of arguments. Create folder "files" then create in it 10 files for each entered argument as extention of files, write in each file current user logged in, and date of creation.

   Example: ~$ ./make_files doc mp3 txt

2. write shell script ***"sort_files"*** take one argument, the path of "files" directory that you create in previous script (or any directory you want to sort its files). The script creates a directory for each extension in the files directory and moves each file to corresponding subdirectory according to its extension.

Note: you don't need to pass extensions as arguments, the script must discover them.

Example: ~$ ./sort_files ./Desktop/Files



Q4- Objective: Write a bash shell script to build a student grade Data Base.

Sample of the output data file

Class Grade Report

Name	OS1	NAP	SE	Average	Level
Ahmad	89	74	74	80.00	B
Maher	81	75	67	75.8	C
Rahaf	88	92	95	91.00	A
Nada	52	69	62	66.66	D

Notes:

1. The average grade for each student is computed as the following

Average= (OS1*2.0+NAP*2.0+SE*1.0)/5

- Success grade=60 ,fail grades must not be calculated in average, and their waight must be deleted:

For example: Nada average=(69*2.0+62*1)/3

- Level : 60 | 70 | 80 | 90 | 100
  D | C | B | A

2. The average grade and Level for each student are computed by using **avg** function.

Script Details

Write bash shell script "grades". This script must take one of the following options. The bash script should check if there are an option provided, if option is exist and it is one of avaliable options, then call corresponding function, otherwise display message to show available options and usage to user.

1. –C : Construct_data_file

Usage: grades –C *outputDataFile*

"**construct_data_file**" function takes one argument, Th name of the output report file to be saved, such as grades.txt. If the outputfile exists then show warning message to overwrite it, else create it.

1. Function asks if you want to enter a student record, you may answer "YES" or "NO".

A "yes¨ response will prompt you to enter student's name, then it will loop the subjects and ask you to enter the corresponding grade for each subject. The script needs to validate the entered score (0<=score<=100)

2. When all the grades (OS1,NAP,SE) for this student are entered, the student average grade should be displayed to the screen (computed by "**avg**" function). Meantime, the student information and the computed grade and Level letter will be recorded to the output file, such as grade.txt

3. A "NO¨ answer will stop recording students grade information, display the current records in the data file to the screen, and exit the bash script.

- The "Avg" function read the new entered student's record (all subjects's grades) and return the average grade and Level letter to the "**construct_data_file**" function.

Note: you can do the following to get the grade from an "Avg" function: grade=`echo $line |Avg`, the grade should be like "96 B". Then, you can use the variable expansion of substrings, such as % and # to get the average score and Level letter. Or you can use the "cut" utility or other UNIX program to do such.

2. –A :add_student

Usage: grades –A *DataFile*

In case the data file exists, Ask you to enter student's name, check if the name exists then show error message, else it will loop to enter subjects's grades. The script needs to validate the entered score (0<=score<=100). Calculate average and Level letter (by using "Avg" function) and finally ***append*** student record to grades.txt.

3. -d :del_student

Usage: grades –d *DataFile std_name*

This function check if such *std_name* exists in the data file. If it exits, it shoud delete the student's record, else show "student is not exist" message.

4. -m :modify_student

Usage: grades –m *DataFile*

The bash script should allow user to modify existing student's record. In case the data file exists, after user entered student name, this function will check if such name exists in the data file. If it exists, it should display the grade during the grade input loop; ask user if it needs to be modified. At the end of input for this student, the existing record will be replaced by the newly input one.

5. –top : top_n_students

Usage: grades –top *DataFile sub_name n*

The bash script allow the teacher to make some Statistics. This function takes three arguments: data file, subject name *sub_name*(OS1,NAP,SE or Average) and number *n,* then it displays the top n students in the required subject and save records to file named Top_n_subject (for example Top_10_OS1).

- ***Note***: Remember your two best friends are the man tool and Google ☺

Administrative Issues:

1. A group may contain **4 students at least and 5 students at most**.
2. The more the script clear and interactive with the user the more will be the solution marks.
3. Student must explain the work of each step (by placing comments within the program).
4. Submission deadline **is 2/1/2014** 12:00 PM, send mail to: network.homeworks@gmail.com
5. Final **Quiz** and discussion on Sunday **5/1/2014**

19/12/2013

With my Best Regards

Eng. Moustafa Najm

mmnajm@gmail.com